Orchestrating a brighter world

NEC

大阪大学
OSAKA UNIVERSITY

# On the Use of Large Language Models for Table Tasks

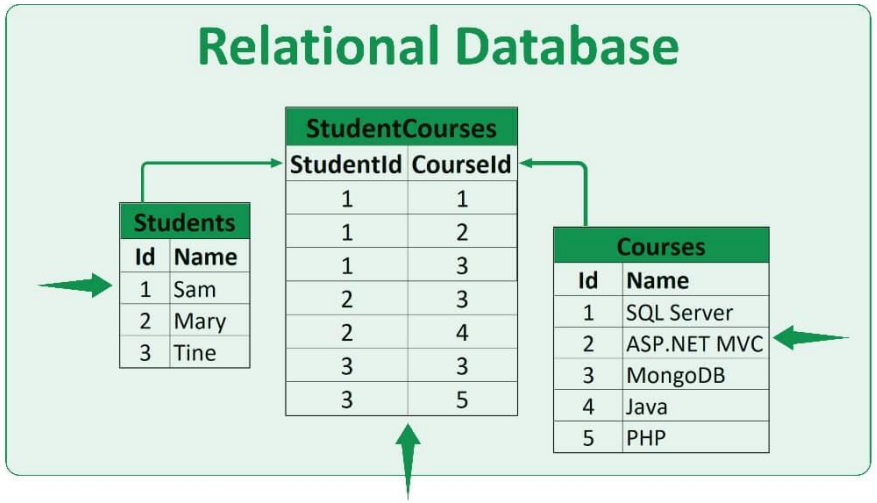
## - Introduction


Haochen Zhang

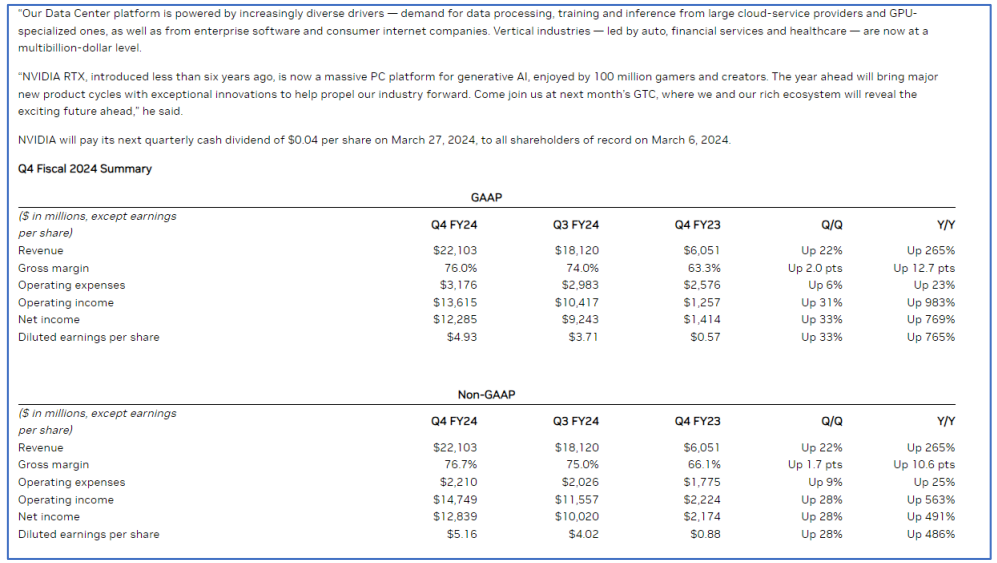
Chuan Xiao

Yuyang Dong

# Tabular data is everywhere




(a)Relational databases



"Our Data Center platform is powered by increasingly diverse drivers — demand for data processing, training and inference from large cloud-service providers and GPU-specialized ones, as well as from enterprise software and consumer internet companies. Vertical industries — led by auto, financial services and healthcare — are now at a multibillion-dollar level.

"NVIDIA RTX, introduced less than six years ago, is now a massive PC platform for generative AI, enjoyed by 100 million gamers and creators. The year ahead will bring major new product cycles with exceptional innovations to help propel our industry forward. Come join us at next month's GTC, where we and our rich ecosystem will reveal the exciting future ahead," he said.

NVIDIA will pay its next quarterly cash dividend of $0.04 per share on March 27, 2024, to all shareholders of record on March 6, 2024.

**Q4 Fiscal 2024 Summary**

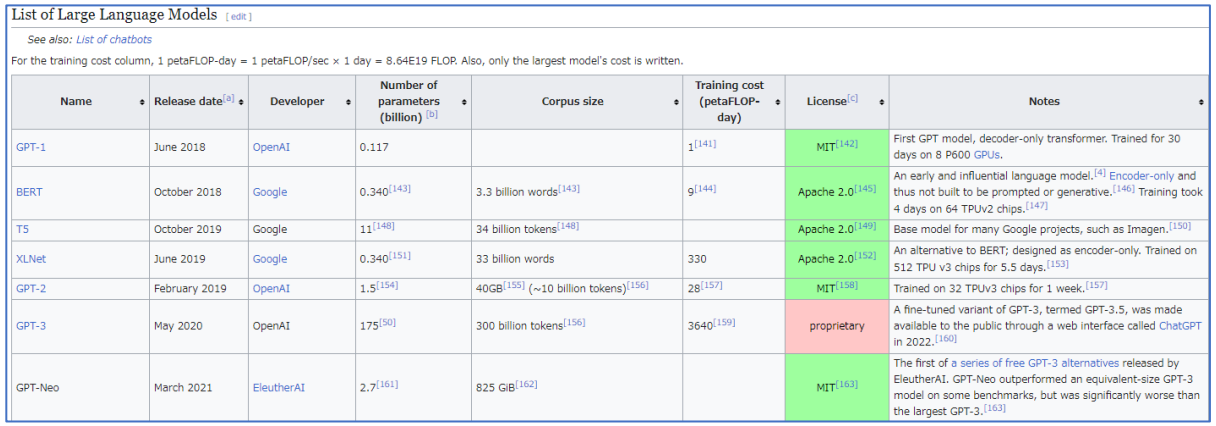
GAAP

| *($ in millions, except earnings per share)* | Q4 FY24 | Q3 FY24 | Q4 FY23 | Q/Q | Y/Y |
|---|---|---|---|---|---|
| Revenue | $22,103 | $18,120 | $6,051 | Up 22% | Up 265% |
| Gross margin | 76.0% | 74.0% | 63.3% | Up 2.0 pts | Up 12.7 pts |
| Operating expenses | $3,176 | $2,983 | $2,576 | Up 6% | Up 23% |
| Operating income | $13,615 | $10,417 | $1,257 | Up 31% | Up 983% |
| Net income | $12,285 | $9,243 | $1,414 | Up 33% | Up 769% |
| Diluted earnings per share | $4.93 | $3.71 | $0.57 | Up 33% | Up 765% |

Non-GAAP

| *($ in millions, except earnings per share)* | Q4 FY24 | Q3 FY24 | Q4 FY23 | Q/Q | Y/Y |
|---|---|---|---|---|---|
| Revenue | $22,103 | $18,120 | $6,051 | Up 22% | Up 265% |
| Gross margin | 76.7% | 75.0% | 66.1% | Up 1.7 pts | Up 10.6 pts |
| Operating expenses | $2,210 | $2,026 | $1,775 | Up 9% | Up 25% |
| Operating income | $14,749 | $11,557 | $2,224 | Up 28% | Up 563% |
| Net income | $12,839 | $10,020 | $2,174 | Up 28% | Up 491% |
| Diluted earnings per share | $5.16 | $4.02 | $0.88 | Up 28% | Up 486% |

(b)Rich documents, PDF



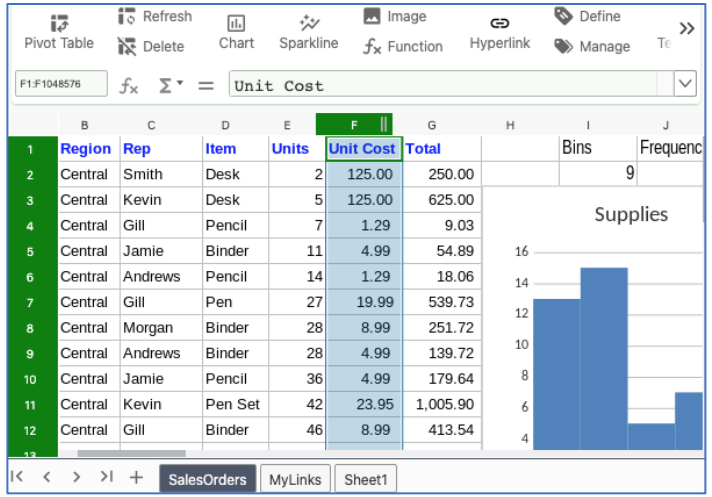
List of Large Language Models [edit]

*See also: List of chatbots*

For the training cost column, 1 petaFLOP-day = 1 petaFLOP/sec × 1 day = 8.64E19 FLOP. Also, only the largest model's cost is written.

| Name | Release date[a] | Developer | Number of parameters (billion) [b] | Corpus size | Training cost (petaFLOP-day) | License[c] | Notes |
|---|---|---|---|---|---|---|---|
| GPT-1 | June 2018 | OpenAI | 0.117 | | 1[141] | MIT[142] | First GPT model, decoder-only transformer. Trained for 30 days on 8 P600 GPUs. |
| BERT | October 2018 | Google | 0.340[143] | 3.3 billion words[143] | 9[144] | Apache 2.0[145] | An early and influential language model.[4] Encoder-only and thus not built to be prompted or generative.[146] Training took 4 days on 64 TPUv2 chips.[147] |
| T5 | October 2019 | Google | 11[148] | 34 billion tokens[148] | | Apache 2.0[149] | Base model for many Google projects, such as Imagen.[150] |
| XLNet | June 2019 | Google | 0.340[151] | 33 billion words | 330 | Apache 2.0[152] | An alternative to BERT; designed as encoder-only. Trained on 512 TPU v3 chips for 5.5 days.[153] |
| GPT-2 | February 2019 | OpenAI | 1.5[154] | 40GB[155] (~10 billion tokens)[156] | 28[157] | MIT[158] | Trained on 32 TPUv3 chips for 1 week.[157] |
| GPT-3 | May 2020 | OpenAI | 175[50] | 300 billion tokens[156] | 3640[159] | proprietary | A fine-tuned variant of GPT-3, termed GPT-3.5, was made available to the public through a web interface called ChatGPT in 2022.[160] |
| GPT-Neo | March 2021 | EleutherAI | 2.7[161] | 825 GiB[162] | | MIT[163] | The first of a series of free GPT-3 alternatives released by EleutherAI. GPT-Neo outperformed an equivalent-size GPT-3 model on some benchmarks, but was significantly worse than the largest GPT-3.[163] |

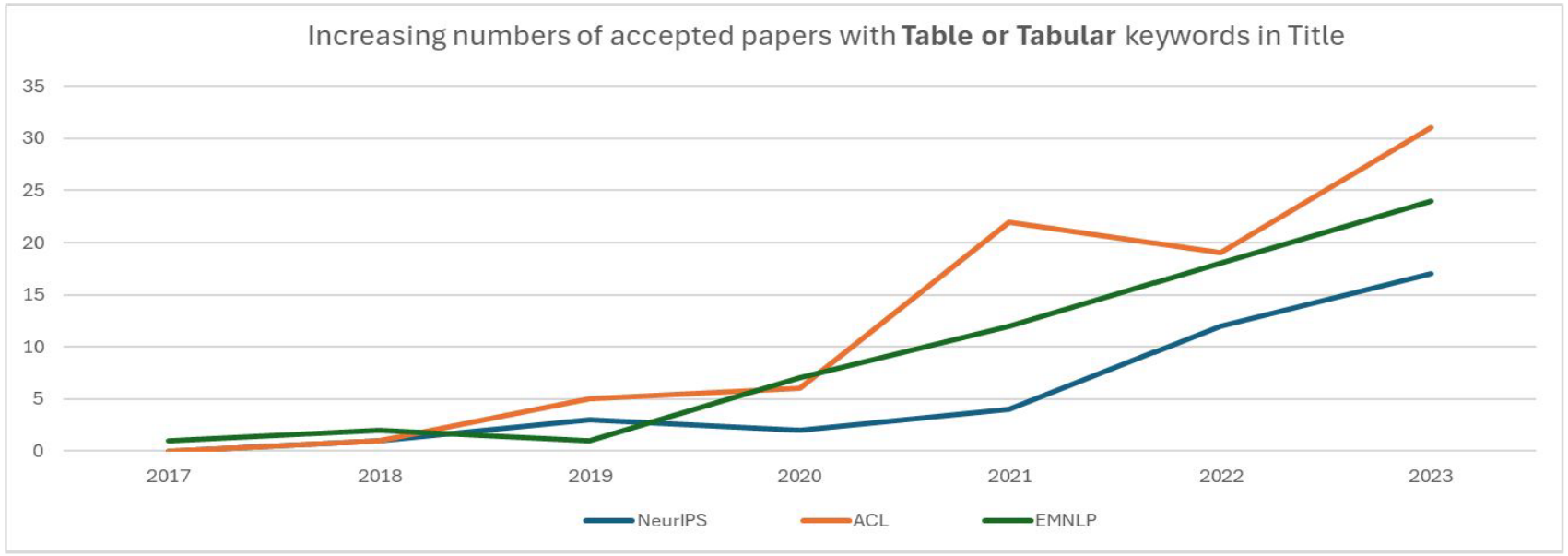
(c) webpages




(d) spreadsheet

# Growing research focus

- Grow quickly in DB, AI and NLP communities




result from [4]

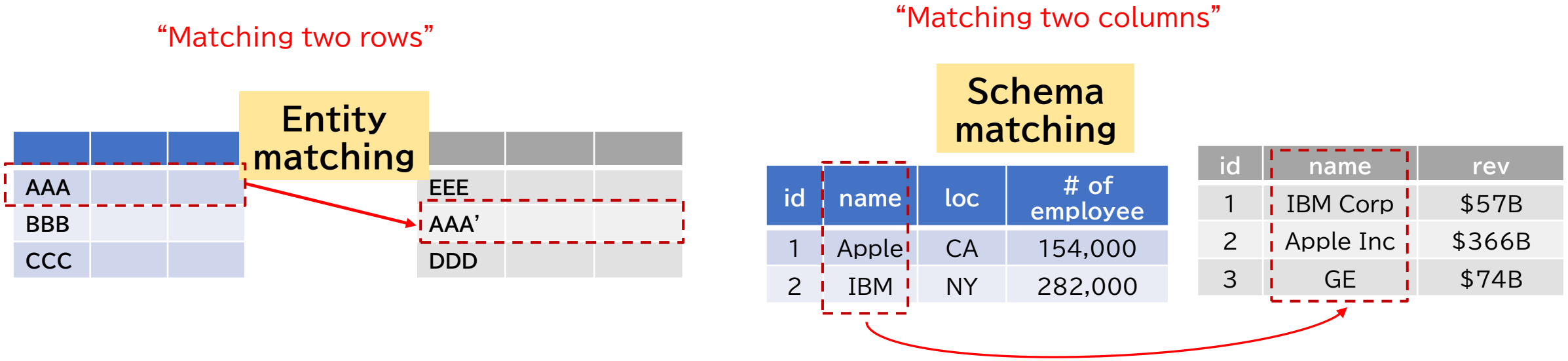
- Recent tutorials
  - [1] Web table extraction, retrieval and augmentation, SIGIR19
  - [2] From Tables to Knowledge: Recent Advances in Table Understanding, KDD21
  - [3] Transformers for Tabular Data Representation: A tutorial on Models and Applications VLDB22, SIGMOD23
  - [4] Large Language Models for Tabular Data: Progresses and Future Directions, SIGIR24
  - A-Paper-List-of-Awesome-Tabular-LLMs, https://github.com/SpursGoZmy/Awesome-Tabular-LLMs

# Table tasks & benchmarks

| “Prepare tables” | “Understand tables” | “Get answer from tables” |
| --- | --- | --- |
| Table **reprocessing** | **Table understanding** | **Table analysis** |
| • Table matching<br>  • Entity matching<br>  • Schema matching<br>• Table cleaning<br>  • Error detection<br>  • Data imputation<br>• Table augmentation<br>  • Row population<br>  • Schema augmentation<br>• Table search<br>• Table transformation | • Table Interpretation<br>  • Entity Linking<br>  • Column Type Annotation<br>  • Relation Extraction<br>• Table detection | • Table QA<br>• Table fact verification<br>• Table-to-text<br>• Text-to-SQL |

# Table matching

Table preprocessing




- **Dataset and benchmark**

1. Can Foundation Models Wrangle Your Data? [VLDB23]. https://arxiv.org/abs/2205.09911
2. Jellyfish: Instruction-Tuning Local Large Language Models for Data Preprocessing [EMNLP24] https://arxiv.org/abs/2312.01678

# Table cleaning

Table preprocessing

"Detect the error cells"

**Error detection**

*inconsistency*

| ID | Name | Date of Birth | Prefecture | Postal Code | Height |
|---|---|---|---|---|---|
| 1 | Yuka | 2003/02/26 | Hokkaido | 540-8570 | 165 |
| 2 | Nana | | Aichi | 464-0804 | 157 |
| 3 | Miho | 2001/06/25 | Kangawa | 2208799 | 1.60 |

*missing* *typo* *format* *outlier*

**Data imputation**

"Impute values into cells"

| ID | Name | Date of Birth | Prefecture | Postal Code | Height |
|---|---|---|---|---|---|
| 1 | Yuka | 2003/02/26 | Osaka | 540-8570 | 165 |
| 2 | Nana | 2003/03/30 | Aichi | 464-0804 | 157 |
| 3 | Miho | 2001/06/25 | Kanagawa | 220-8799 | 160 |

- **Dataset and benchmark**

1. Can Foundation Models Wrangle Your Data? [VLDB23]. https://arxiv.org/abs/2205.09911
2. Jellyfish: Instruction-Tuning Local Large Language Models for Data Preprocessing [EMNLP24] https://arxiv.org/abs/2312.01678

# Table augmentation

Table preprocessing

"Add columns/rows to table"

Column population

| id | name | loc | # of employee | |
|---|---|---|---|---|
| 1 | Apple | CA | 154,000 | |
| 2 | IBM | NY | 282,000 | |
| | | | | |

Row population

- **Dataset and benchmark**

1. TURL: Table Understanding through Representation Learning. [VLDB20] https://arxiv.org/abs/2006.14806
2. TableLlama: Towards Open Large Generalist Models for Tables. [NAACL24]. https://osu-nlp-group.github.io/TableLlama/

# Table search

Table preprocessing

"Retrieve tables with an NL query"

**"Tables contains information of Apple Inc."**

| id | name | loc | # of employee |
|---|---|---|---|
| 1 | Apple | CA | 154,000 |
| 2 | IBM | NY | 282,000 |

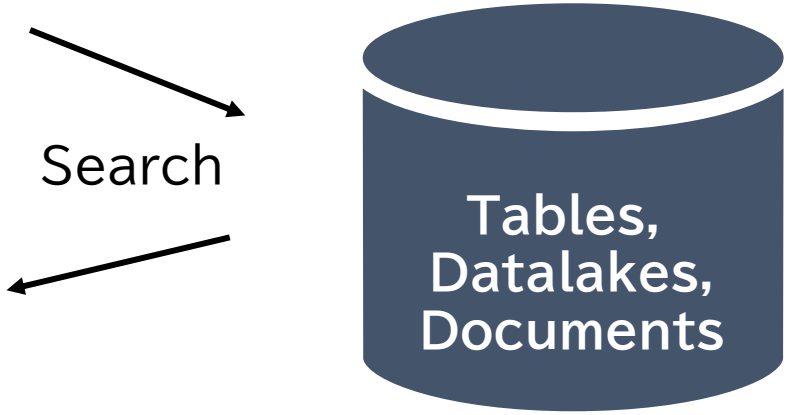



**Table collection**

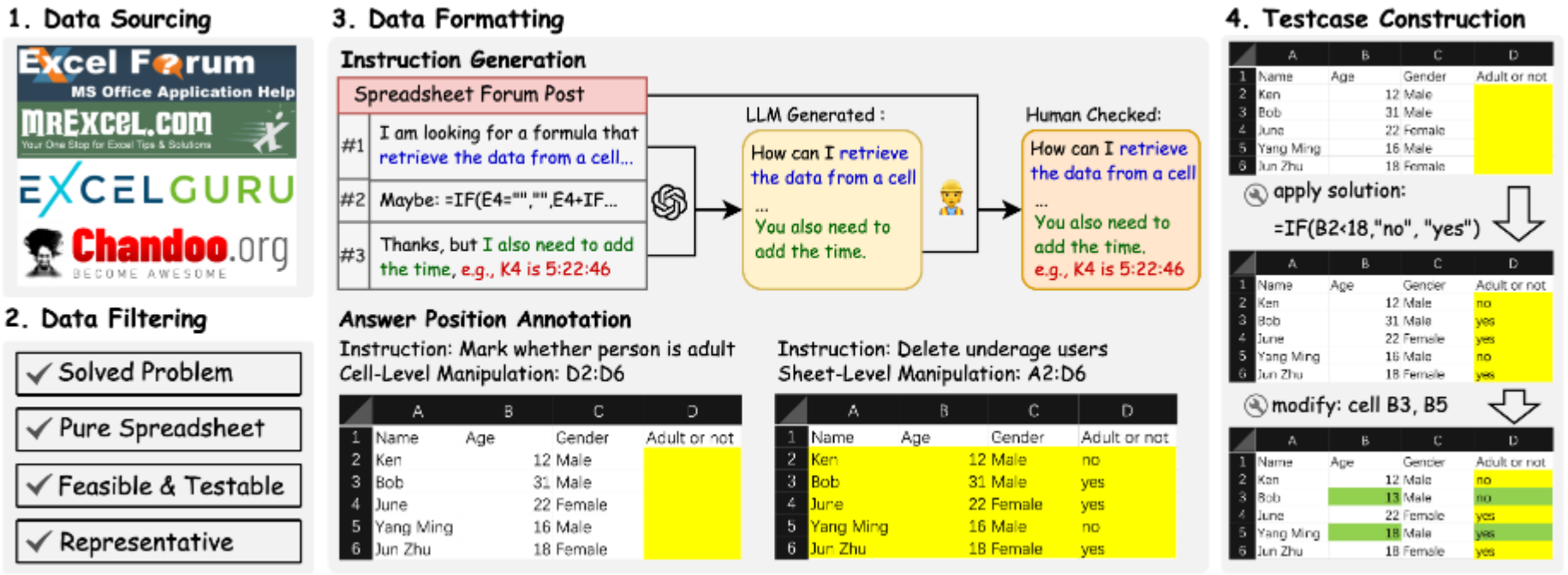
- **Dataset and benchmark**

1. Open-Domain Table Retrieval for Natural Questions. https://github.com/zorazrw/nqt-retrieval
2. Open-WikiTable :Dataset for Open Domain Question Answering with Complex Reasoning over Table [EMNLP23] https://github.com/sean0042/Open WikiTable

# Table transformation

Table preprocessing

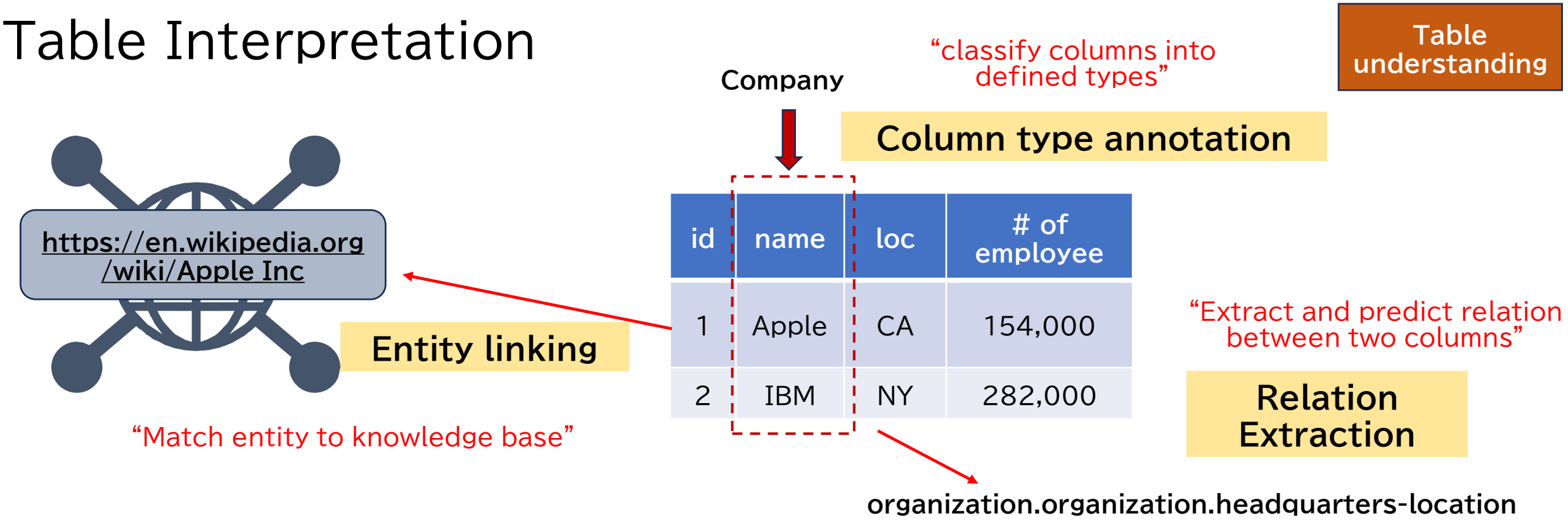
"Manipulate table into wanted styles"




- **Dataset and benchmark**

1. SpreadsheetBench: Towards Challenging Real World Spreadsheet Manipulation [NeurIPS24] https://spreadsheetbench.github.io/

# Table Interpretation




- **Dataset and benchmark**

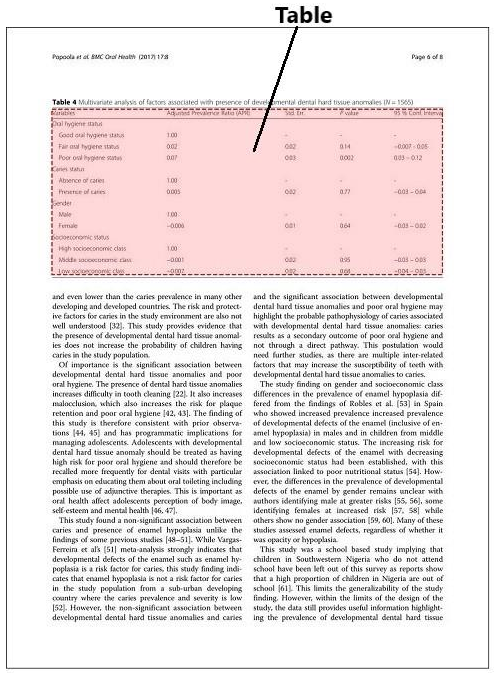
1. TURL: Table Understanding through Representation Learning. [VLDB20] https://arxiv.org/abs/2006.14806
2. TableLlama: Towards Open Large Generalist Models for Tables. [NAACL24]. https://osu-nlp-group.github.io/TableLlama/
3. Column Type Annotation using ChatGPT [arixv24] https://arxiv.org/abs/2306.00745

# Table detection

"detect table region, structure and content"

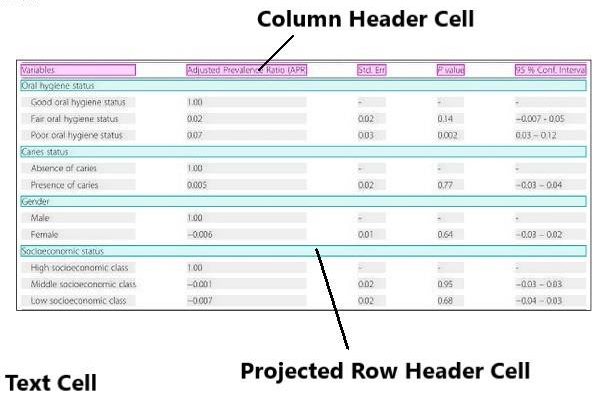
**Table understanding**

### Table Detection




### Table Structure Recognition


### Table Functional Analysis




- **Dataset and benchmark**

1. PubTables-1M: Towards comprehensive table extraction from unstructured.[CVPR22] https://github.com/microsoft/table-transformer
2. TableFormer: Table Structure Understanding with Transformers. [CVPR22]. https://github.com/IBM/TabFormer
3. Docling Technical Report. [arxiv24] https://github.com/DS4SD/docling-ibm-models

# Table QA

Table analysis

"Question-answering on tabular contents"

| id | name | loc | # of employee |
|---|---|---|---|
| 1 | Apple | CA | 154,000 |
| 2 | IBM | NY | 282,000 |

**Question 1: Where is the location of IBM?**
**Answer 1: New York**

**Question 2: What is the sum of employee in Apple and IBM?**
**Answer 2: 436,000**

Table 1: An overview of table QA datasets. The representative methods without marks (e.g. †☆❣) can be used on the datasets aligned in the same horizontal zone, and the methods with marks are currently adopted on the datasets with the same mark.

| Dataset | | Closed -domain | Question Type | Representative Methods |
|---|---|---|---|---|
| Table -only | WTQ☆ [36] | Yes | Factoid | Semantic parsing-based [10,11,12,16,19,20,28,34,36] [38,41,42,44,46,50] Generative method❣ [31] Matching-based method† [15] Extractive method☆ [18] |
| | SQA❣☆ [20] | Yes | Factoid | |
| | WikiSQL☆ [50] | Yes | Factoid | |
| | Spider [47] | Yes | Factoid | |
| | HiTab [8] | Yes | Factoid | |
| | AIT-QA†☆ [23] | Yes | Factoid | |
| Non -table -only | FeTaQA[32] | Yes | Free form | Generative method [32] |
| | FinQA [7] | Yes | Factoid | Semantic parsing-based[7] |
| | TAT-QA [52] | Yes | Factoid | Extractive methods [6,13,52] |
| | HybridQA [6] | Yes | Factoid | |
| | TabMCQ [22] | Yes | Multiple choice | Matching-based methods [22,27] |
| | GeoTSQA [27] | Yes | Multiple choice | |
| | OTTQA [5] | No | Factoid | Retriever-reader-based methods [5,17,25,35,51] |
| | NQ-tables [17] | No | Factoid | |

- **Dataset and benchmark**

1. TableLlama: Towards Open Large Generalist Models for Tables. [NAACL24]. https://osu-nlp-group.github.io/TableLlama/
2. https://huggingface.co/datasets/SpursgoZmy/IFT-Data-For-Tabular-Tasks

# Table fact verification

Table analysis

| id | name | loc | # of employee |
|---|---|---|---|
| 1 | Apple | CA | 154,000 |
| 2 | IBM | NY | 282,000 |

"Verify a given sentences according to table"

"IBM and Apple are U.S. companies." → Entailed

"Apple has more employee than IBM." → Refuted

- **Dataset and benchmark**

1. TabFact: A Large-scale Dataset for Table-based Fact Verification.[ICLR20] https://tabfact.github.io/
2. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information. [ACL21 workshop]. https://fever.ai/dataset/feverous.html

# Table-to-text

Table analysis

"Generate description of table"

| id | name | loc | # of employee |
|---|---|---|---|
| 1 | Apple | CA | 154,000 |
| 2 | IBM | NY | 282,000 |

**"The table presents information about two major companies, Apple and IBM, along with their locations and employee counts. Apple, headquartered in California (CA), employs 154,000 people. On the other hand, IBM, based in New York (NY), has a significantly larger workforce, with 282,000 employees."**

- **Dataset and benchmark**

1. Neural Text Generation from Structured Data with Application to the Biography Domain [EMNLP16] https://github.com/DavidGrangier/wikipedia-biography-dataset
2. ToTTo: A Controlled Table-To-Text Generation Dataset [EMNLP20] https://huggingface.co/datasets/google-research-datasets/totto
3. Table-to-text: Describing table region with natural language. [AAAI18] https://github.com/msra-nlc/Table2Text

# Text-to-SQL

Table analysis

"Convert natural language to SQL query"

**Text: How many employees in Apple?**

```
SELECT `employee_num`
FROM table_name
WHERE name = 'Apple';
```

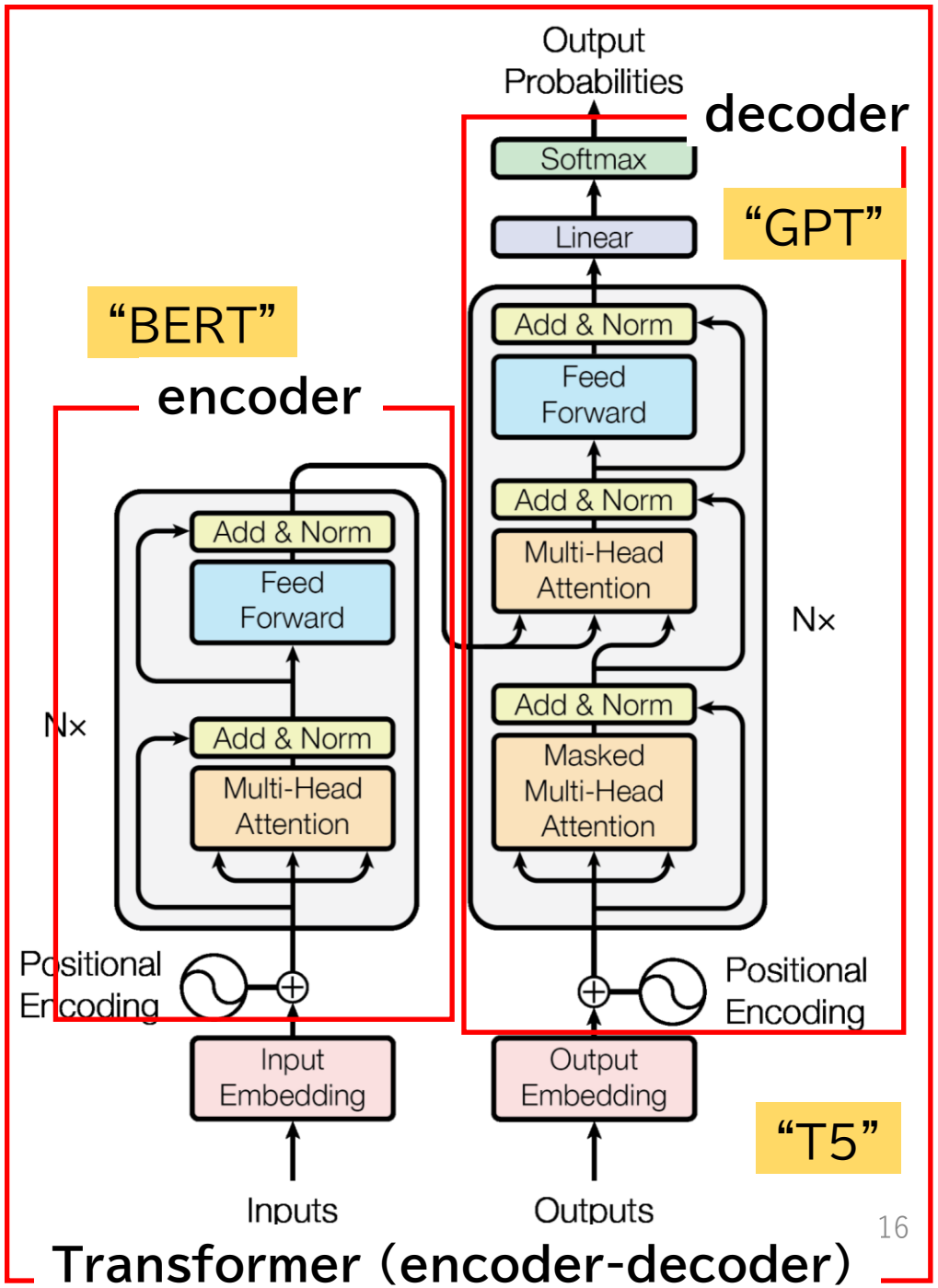
| id | name | loc | employee_num |
|---|---|---|---|
| 1 | Apple | CA | 154,000 |
| 2 | IBM | NY | 282,000 |

- **Dataset and benchmark**

1. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning.[ICLR18] https://github.com/salesforce/WikiSQL
2. Spider [EMNLP18] https://yale-lily.github.io/spider

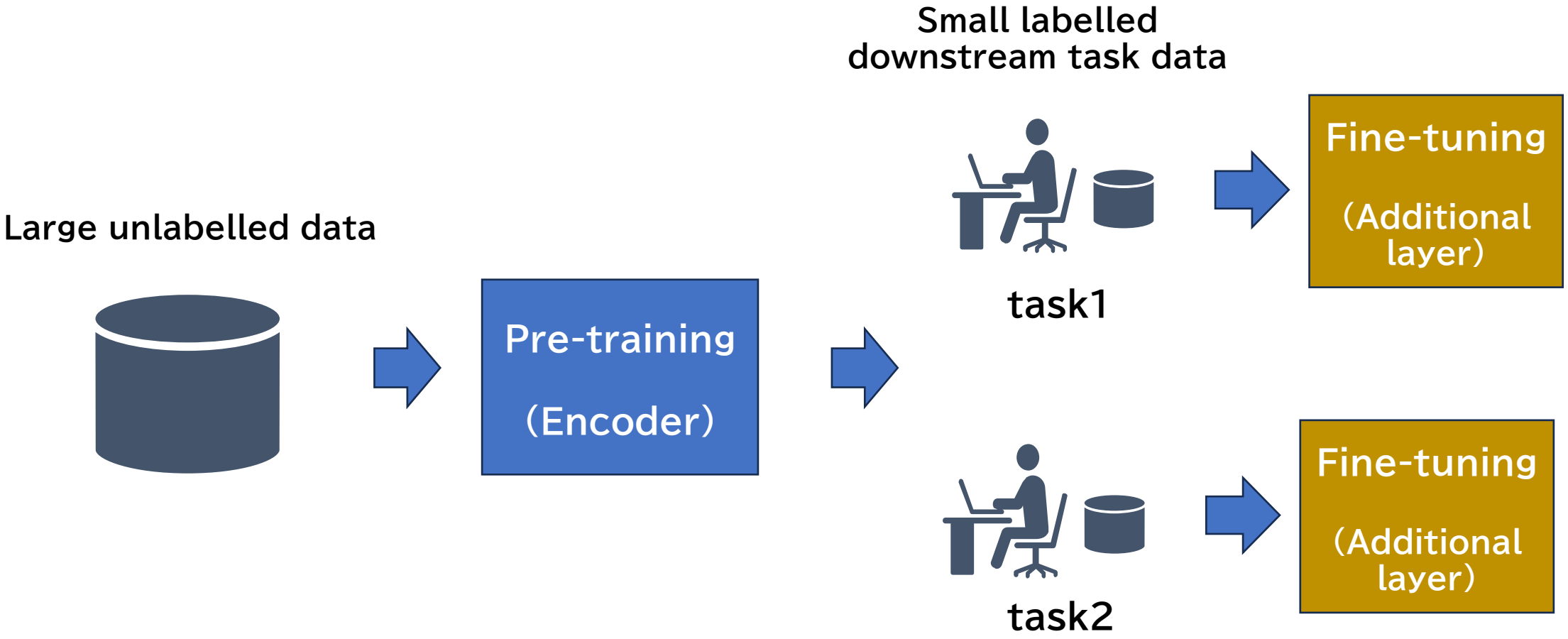
# Methods before LLM

- Rule, ML, NN-based -> skip
- Transformer-based (2018-)
  - Encoder
  - Encoder-Decoder



# Motivation of Encoder for tables

- Pretrain-and-finetune (“BERT-way”)
  - Learning good table representation (embedding) with table pretraining tasks
  - Finetune on downstream tasks



# Table-only Pretraining

- Pretrain with table contents

- TURL [VLDB20]
  - Masked Language Model (MLM)
  - Masked Entity Recovery

- TABBIE [NAACL21]
  - Detect corrupted cells

- TUTA [KDD21]
  - MLM
  - Cell filling
  - Context selection

TURL

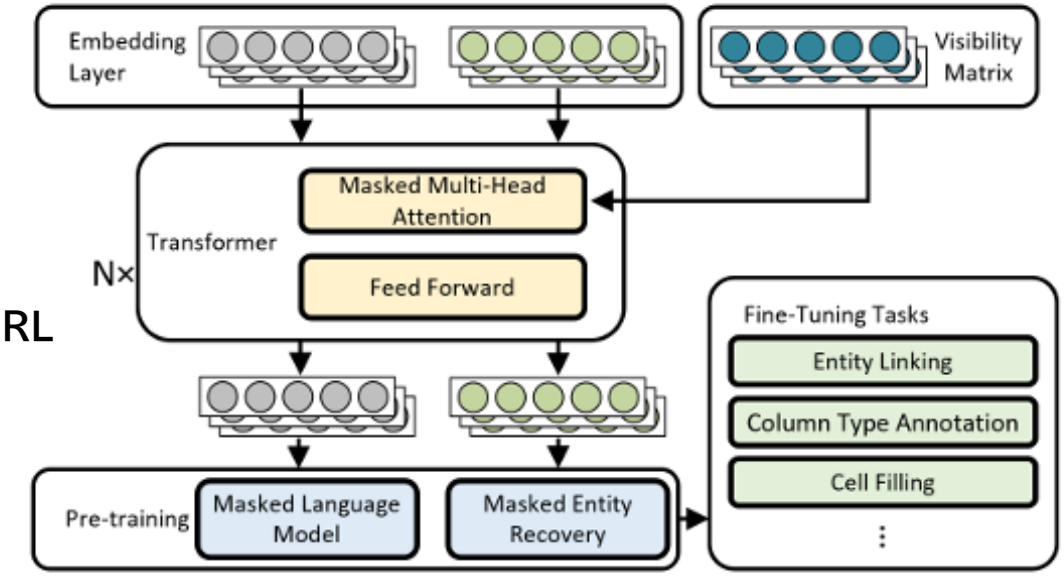
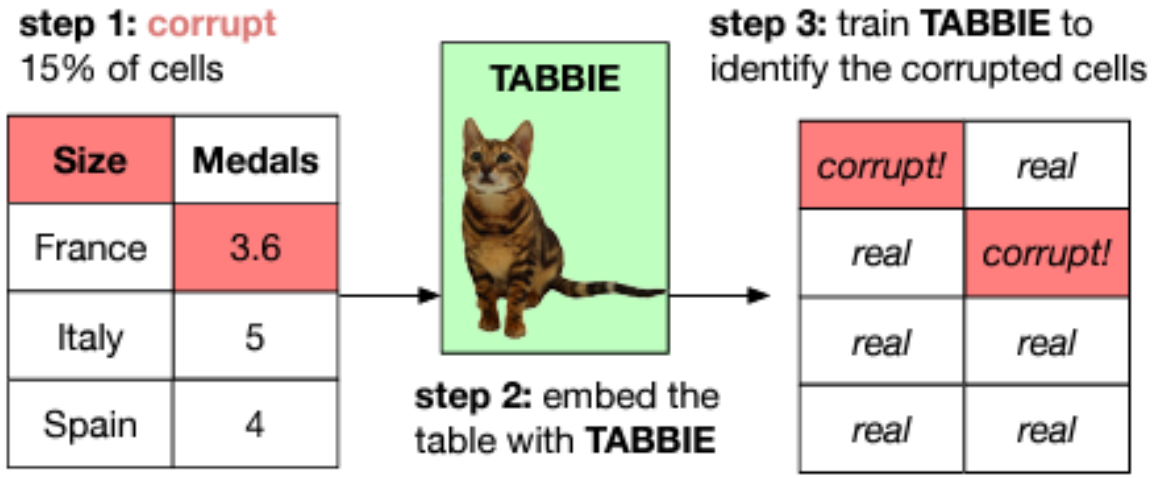



**Figure 2: Overview of our TURL framework.**

TABBIE



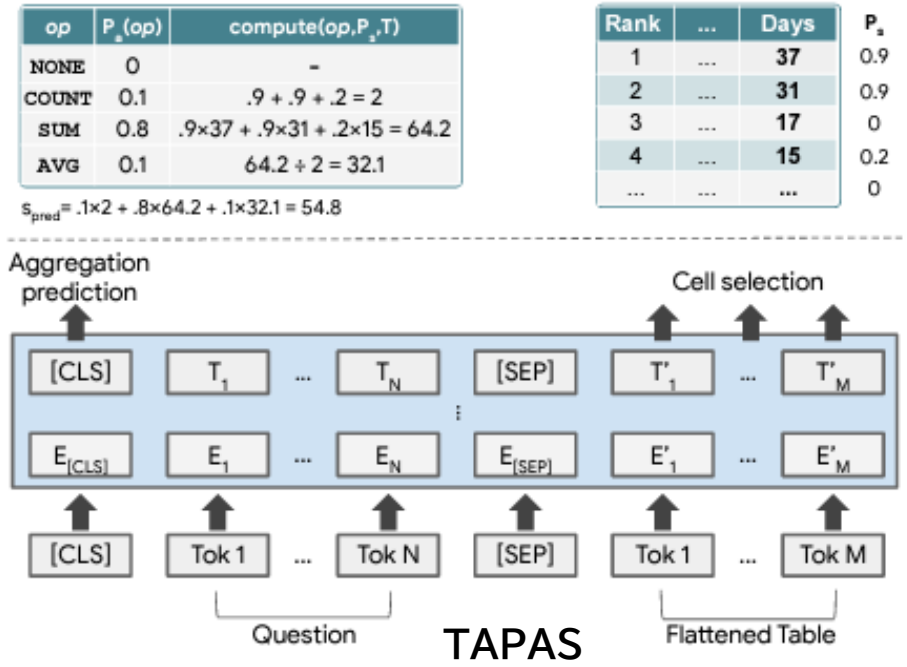

TURL: https://arxiv.org/pdf/2006.14806
TABBIE: https://arxiv.org/abs/2105.02584
TUTA: https://arxiv.org/pdf/2010.12537

# Table-and-query Pretraining

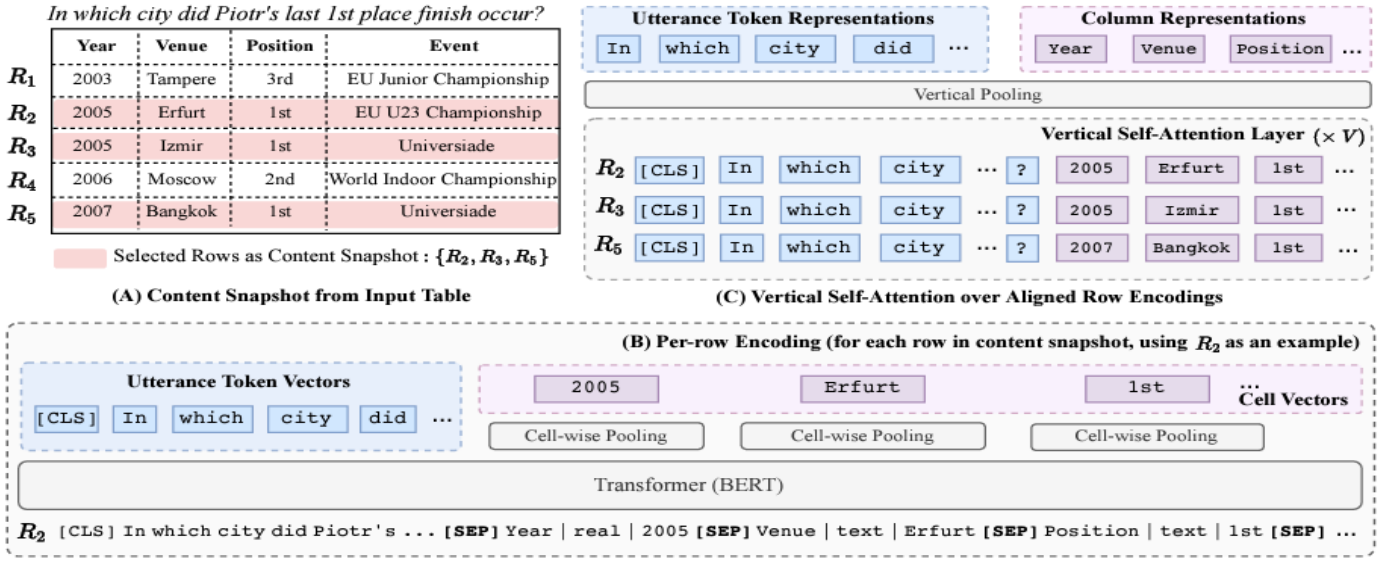
- Pretrain with table contents & query

- TAPAS[ACL20]
  - Query and whole table
    - Aggregation prediction
    - Cell selection task

- TaBERT[NAACL21]
  - Query and related rows
    - Masked Language Model




TAPAS

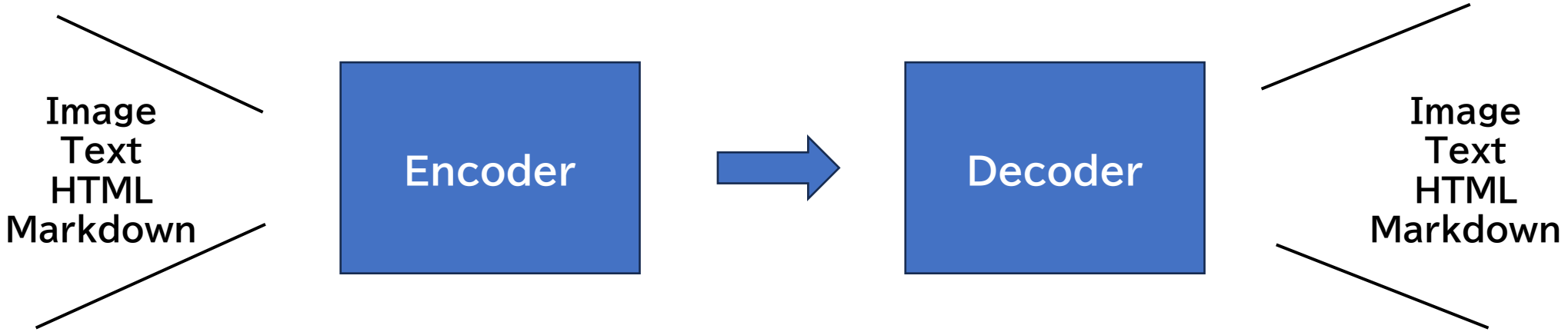



TaBERT

TAPAS: https://arxiv.org/abs/2004.02349
TaBERT: https://arxiv.org/abs/2005.08314

# Motivation of Encoder-decoder for tables

- Flexible input and output
  - Table to text
  - Text to sql
  - Table summarize
  - Table to markdown, html
- Mulitmodal ability
  - Image encoder -> text decoder: table OCR task, table VQA
- Generalized and good generation ability



# Text-to-text Encoder Decoder

- Generalized and Good generative ability by fine-tuning on pretrained encoder decoder model

- UnifiedSKG [EMNLP22]
  - Fine tune T5

- TaPEx [ICLR22]
  - Fine tune BART

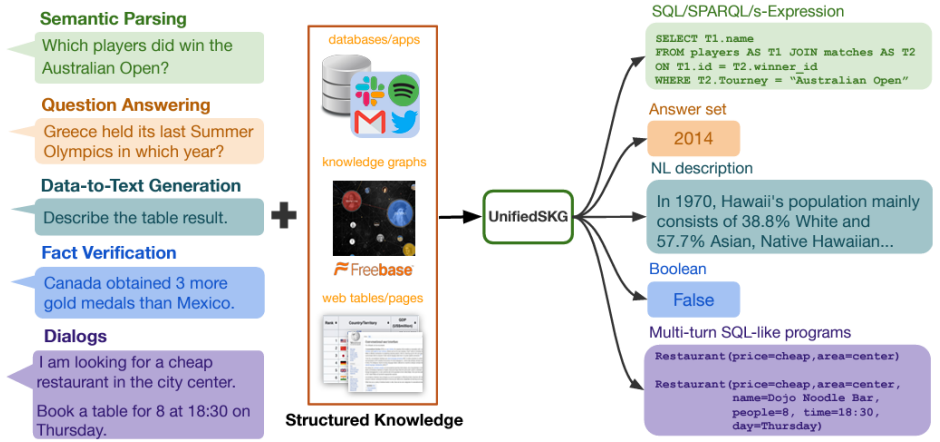
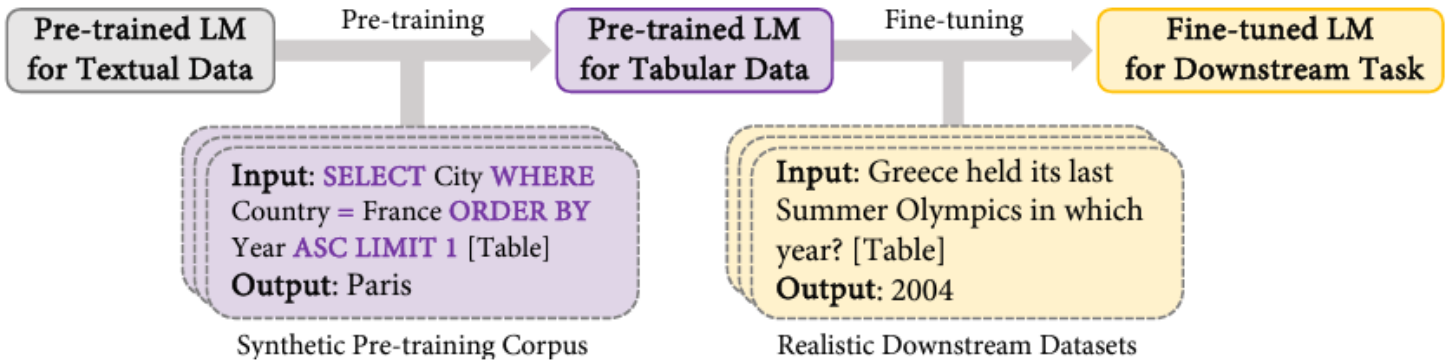



Figure 1: Structured knowledge grounding (SKG) leverages structured knowledge to complete user requests. By casting inputs and outputs into the text-to-text format, UNIFIEDSKG standardizes datasets, models, code, experiments, and metrics for 21 SKG tasks.

UnifiedSKG

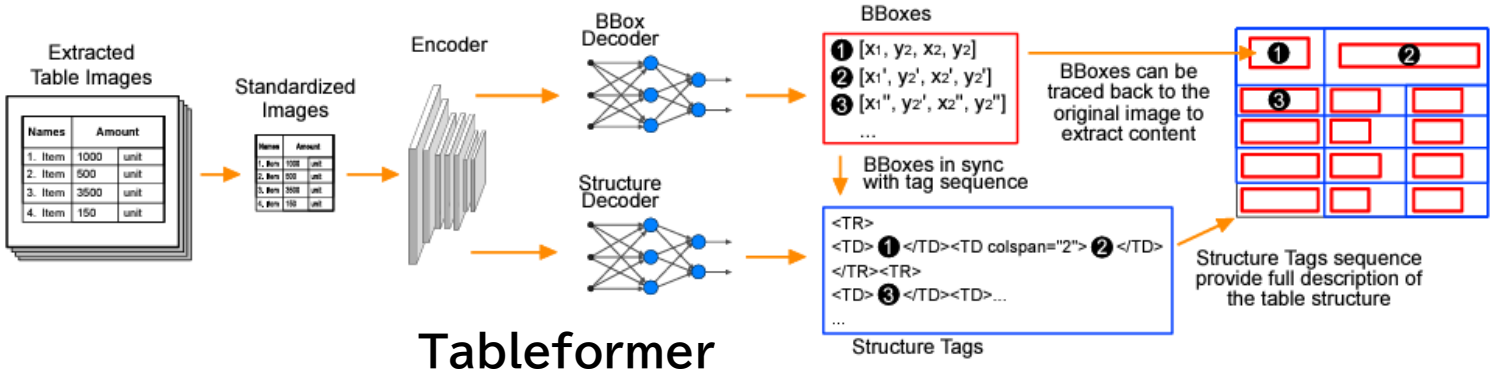



TaPEx

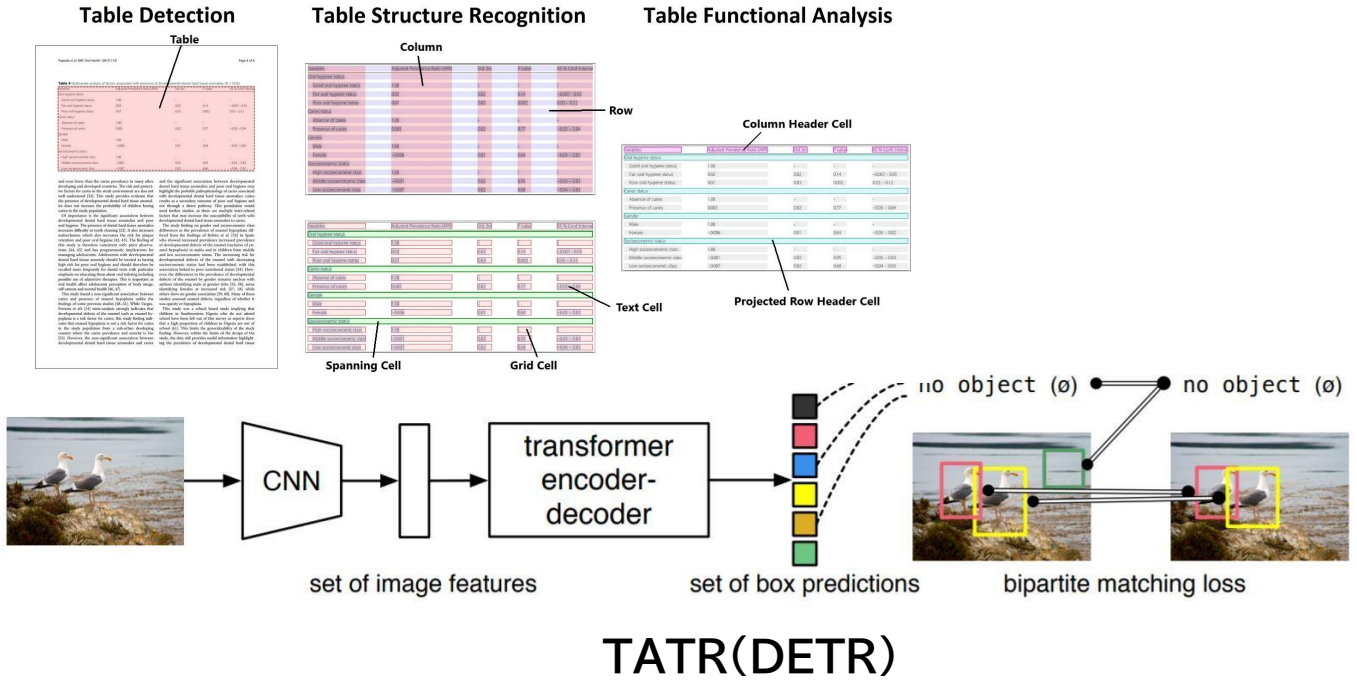
UnifiedSKG: https://arxiv.org/abs/2201.05966
TaPEx: https://arxiv.org/abs/2107.07653

# Vision Encoder Decoder

- Multimodality
- Tableformer [CVPR22]: Table detection & OCR
  - Bounding box detection, structure generation



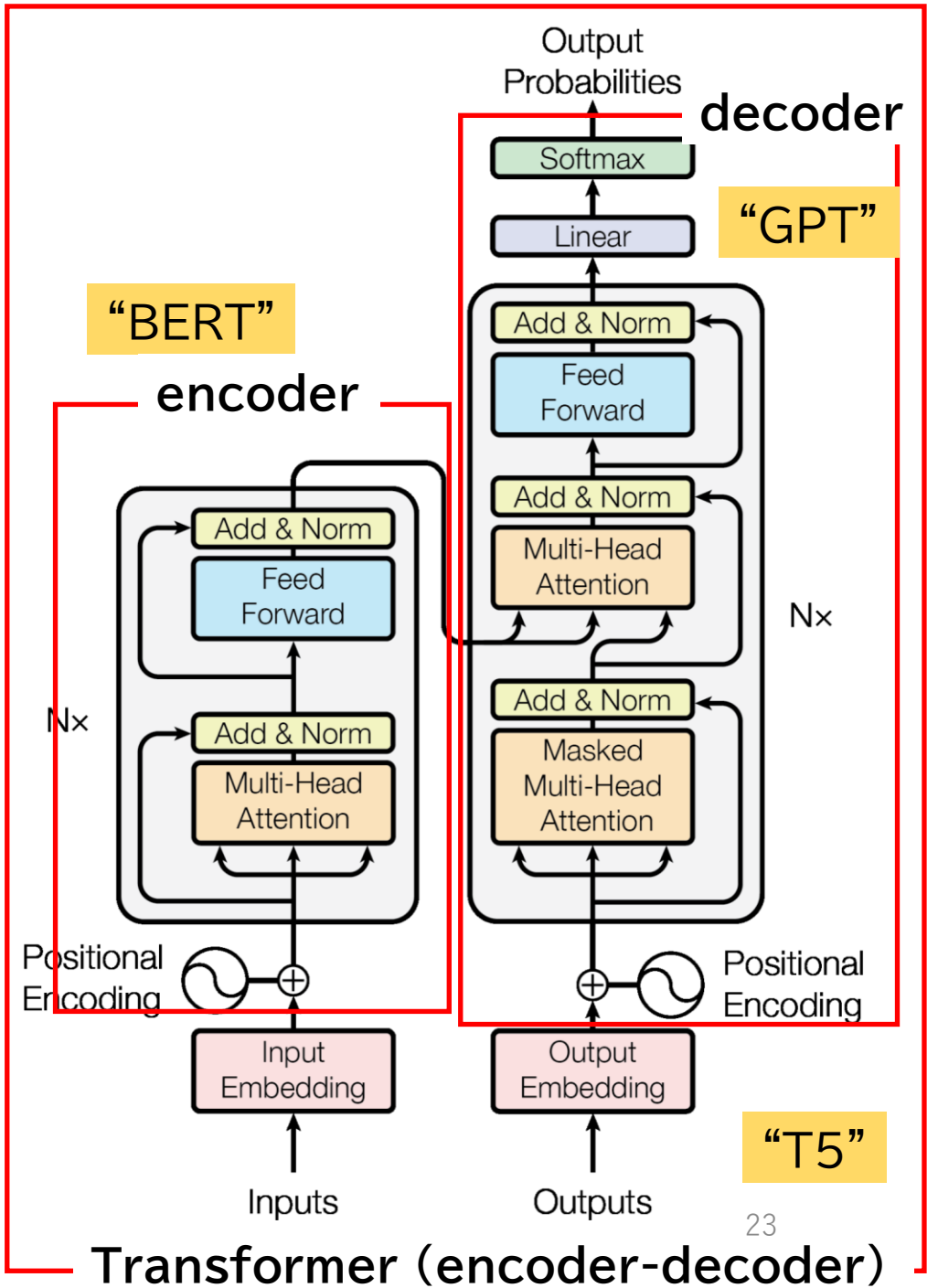

- TATR [CVPR22]
  - Table detection
  - Based on object detection transformer (DETR)




Tableformer: https://arxiv.org/abs/2203.01017
TATR: https://github.com/microsoft/table-transformer

# LLM (decoders for table)

- Autoregressive Generation
  - Good for generative task
  - Easy for self-supervised training
- Simple architecture
  - Efficient on training large-scale data
  - Easy to scale the model



# Motivation of LLM for table task

- Good points
  - Large knowledge
  - In context learning/ zero shot ability
  - Generation task, interactive

- Limitation
  - Slow
  - Cost
  - Not good performance for
    - Mathematical task
    - Large table

# Use of LLM in table tasks

Prompt

Fine-tuning

RAG

LLM agents

VLM

Orchestrating a brighter world

NEC

大阪大学
OSAKA UNIVERSITY

# On the Use of Large Language Models for Table Tasks

- Prompting

# Prompt Engineering

## Zero-shot

- Instruct the LLM to follow prompts.

## Few-shot

- Demonstrate with one or more examples.

## Chain-of-thought

- Elicit complex reasoning through a series of intermediate reasoning steps.
- Variants: **tree-of-thought**, **graph-of-thought**, etc.

### Bad Prompt

I'm hungry.

Certainly, here's a whole bunch of useless text because I don't know what you want because I can't read your mind because all I do is generate text from prompts. If you are unsatisfied with my response please provide a more useful prompt.

### Good Prompt

I'm an adult man weighing 200 pounds with a daily caloric requirement of 2500. I need to hit my protein goal for the day which is 120 grams. I'm hungry for something hot and a little salty.

Scramble and cook 4 eggs.

source: https://medium.com/the-augmented-engineer/new-to-ai-prompt-writing-learn-how-to-get-summaries-that-cover-all-the-key-points-you-care-about-46615fab135b

# Zero-Shot Prompting

- Instruct the model to follow specific prompts.

I went to the market and bought 10 apples. I gave two to my neighbor and two to the repairman. Then I bought 5 apples and ate one. How many apples are left?
Let's think about this step by step.

First, I bought 10 apples.

1. I gave two apples to my neighbor; 10 - 2 = 8 apples left.
2. I gave two apples to the repairman; 8 - 2 = 6 apples left.
3. Then I bought 5 apples. 6 + 5 = 11 apples left.
4. One apple eaten; 11 - 1 = 10 apples left.

As a result, there are 10 apples left.

Kojima et al. Large language models are zero-shot reasoners. NeurIPS 2022.



source: https://qiita.com/Jazuma/items/9de7c56fd9fa8b9c1648

# Few-Shot Prompting

- Demonstrate task-solving with examples provided in prompts.

Fantastic! // Positive

Terrible! // Negative

The movie was amazing! // Positive

What a horrible show! //

Negative

source: https://qiita.com/Jazuma/items/9de7c56fd9fa8b9c1648

Brown et al. Language models are few-shot learners. NeurIPS 2020.

# Chain-of-Thought

- Elicit complex reasoning by providing inference processes.

The odd numbers in this group add up to an even number. : 4, 8, 9, 15, 12, 2, 1.

A: Adding all odd numbers gives 9+15+1 = 25. The answer is False.

The odd numbers in this group add up to an even number. : 15, 32, 5, 13, 82, 7, 1.

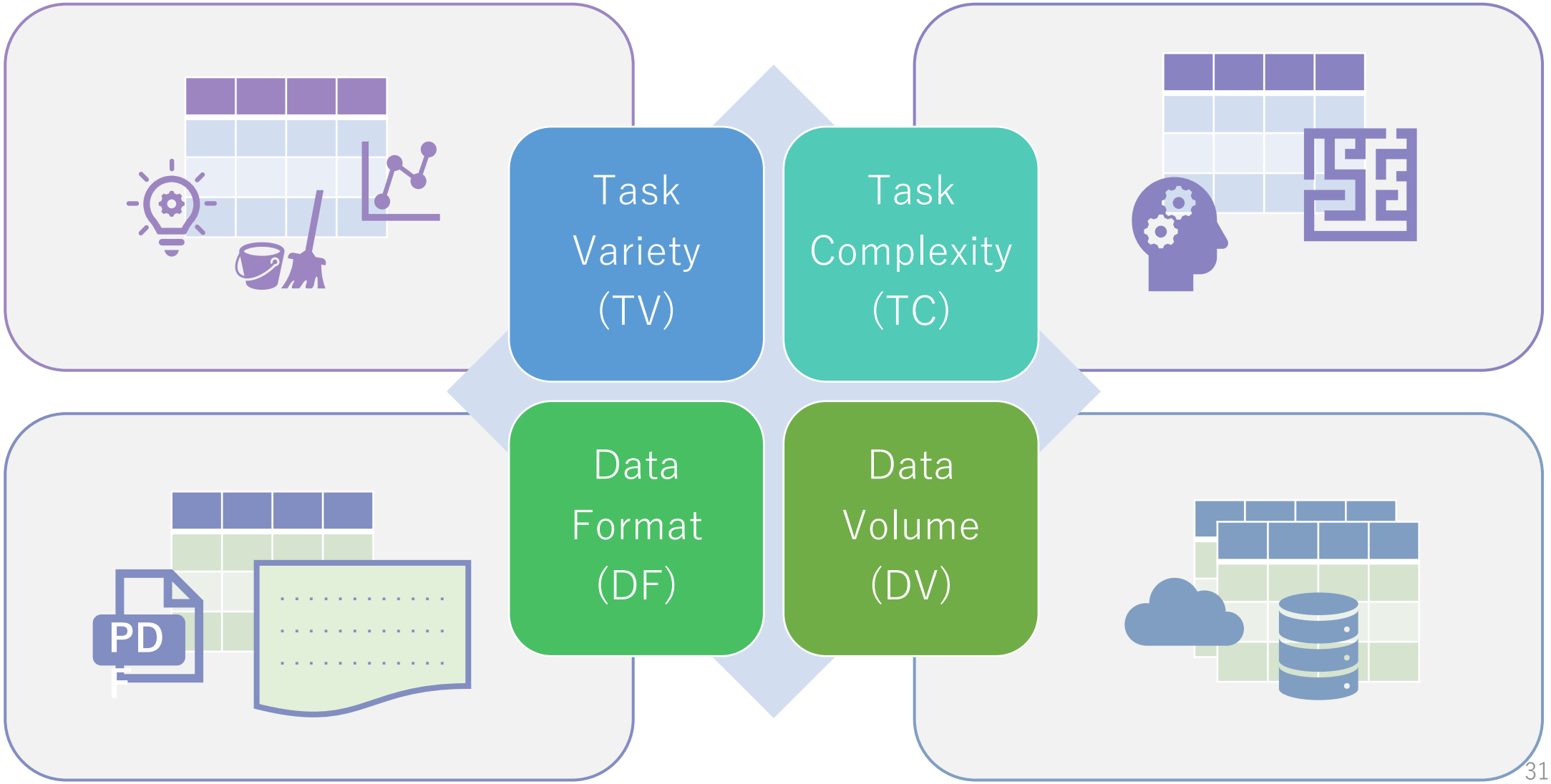
A:

Adding all the odd numbers gives 15 + 5 + 13 + 7 + 1 = 41. The answer is False.

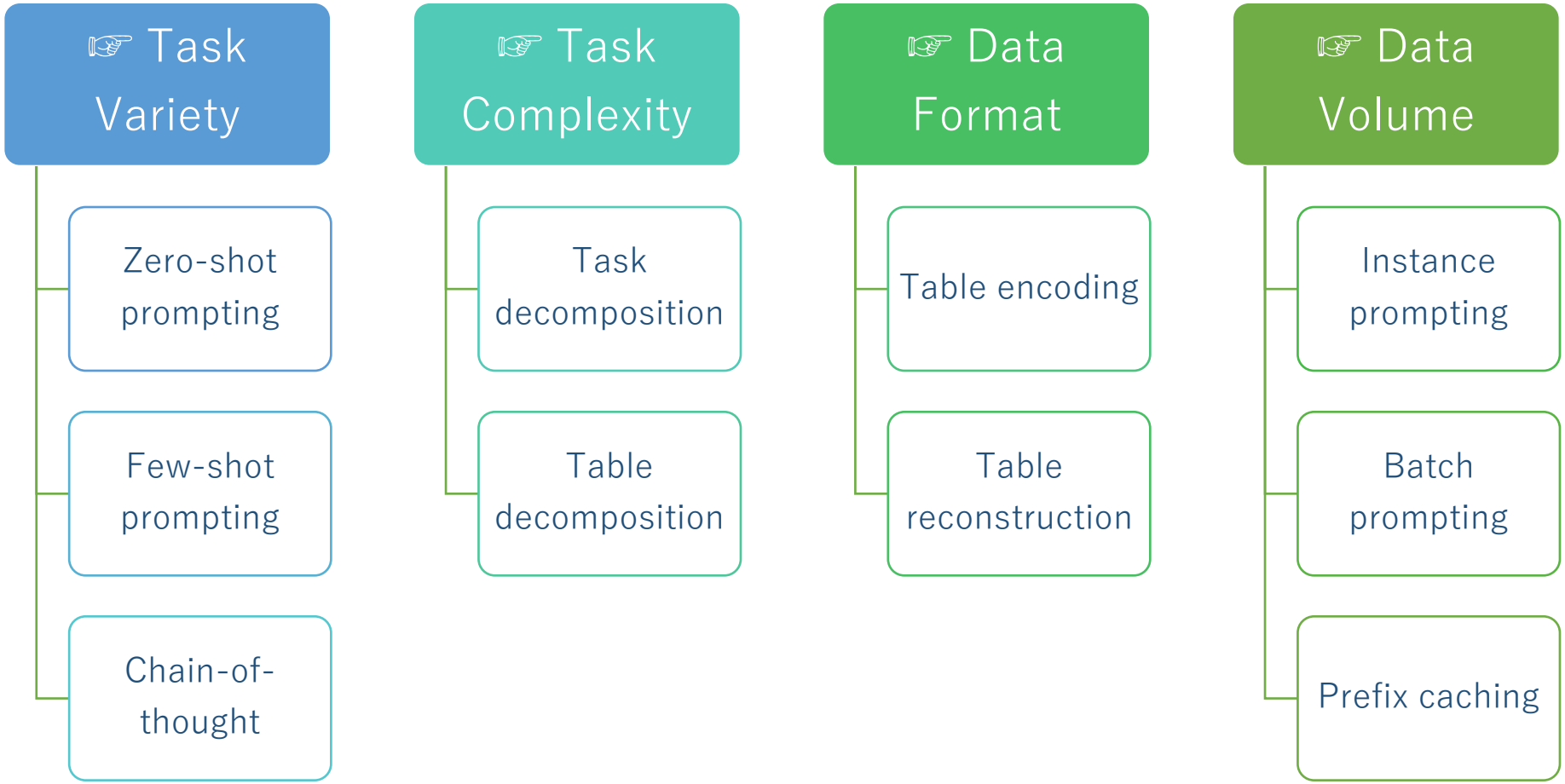
source: https://qiita.com/Jazuma/items/9de7c56fd9fa8b9c1648

Wei et al. Chain of thought prompting elicits reasoning in large language models. NeurIPS 2022.
Wang et al. Self-consistency improves chain of thought reasoning in language models. ICLR 2023.

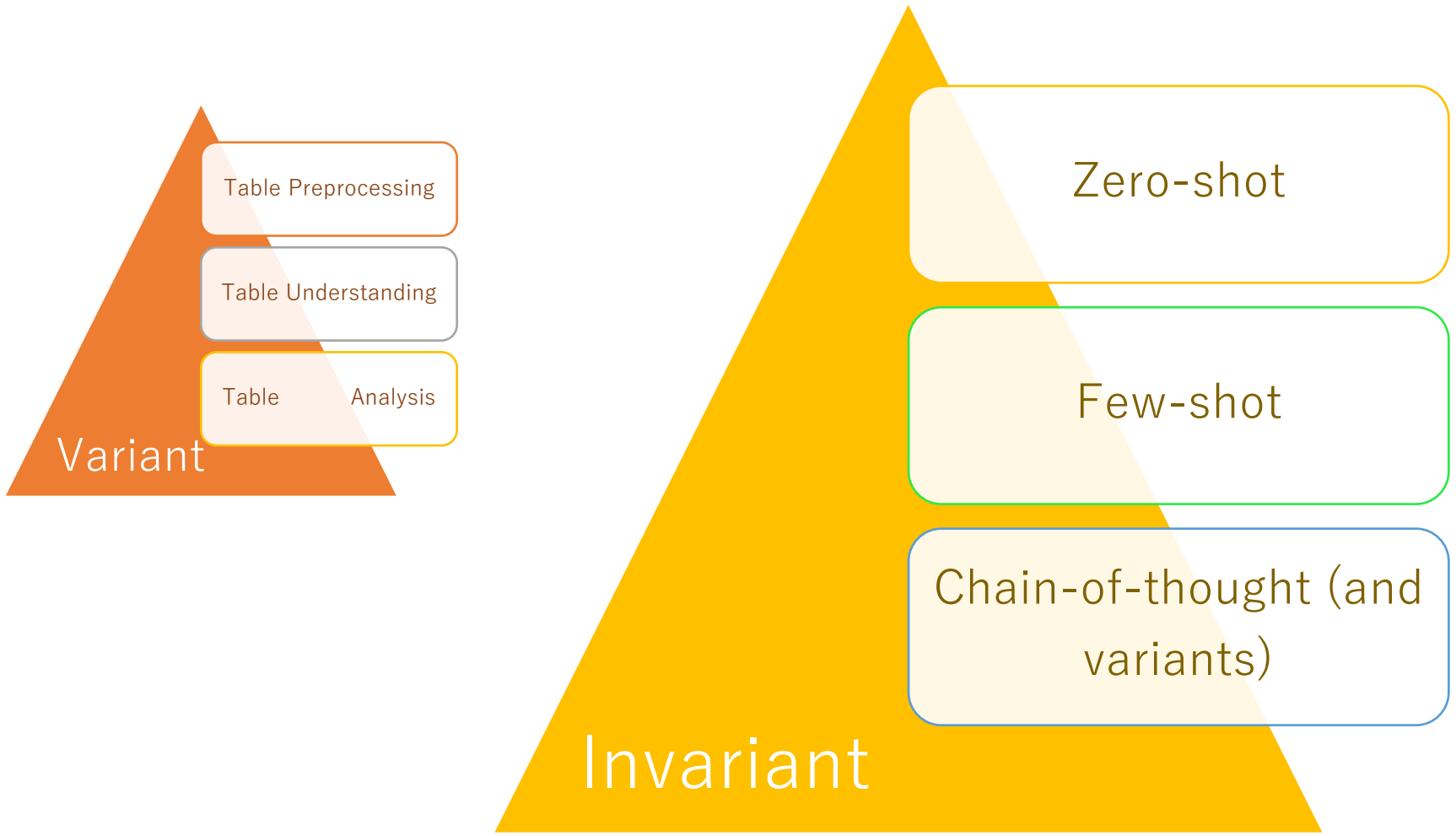
# Issues of Prompting for Table Tasks



# Prompting Techniques for Table Tasks



# ☞ Task Variety



# ☞ Task Variety: Zero-Shot Prompting

| name | city | addr | phone | Type |
|---|---|---|---|---|
| Langer's | ? | 704 S. Alvarado St. | 213-483-8050 | delis |

| Task Type | Data Imputation |
|---|---|
| Task Description | You are a database engineer.<br><br>You are requested to **infer the value of the "city" attribute** based on the values of other attributes. |
| Data Instance | [name: "langer's", addr: "704 s. alvarado st.", phone: "213-483-8050", type: "delis"] |

*prompt*

| Answer | The city is "Los Angeles". |
|---|---|

Zhang et al. Large language models as data preprocessors. TaDA 2024.

Table Preprocessing

# ☞ Task Variety: Few-Shot Prompting

| Task Type | Data Imputation |
|---|---|
| **Task Description** | You are a database engineer.<br><br>You are requested to infer the value of the "city" attribute based on the values of other attributes. |
| **Data Instance** | [name: "langer's", addr: "704 s. alvarado st.", phone: "213-483-8050", type: "delis"] |
| **Few-shot Examples** | **Some examples are given below.**<br>```<br>User:<br>Question 1: Record is [name: "carey's corner", addr: "1215 powers ferry rd.", phone: "770-933-0909", type: "hamburgers"]. What is the city?<br>Assistant:<br>Answer 1: Marietta<br>...<br>``` |
| **Answer** | Los Angeles |

*prompt*

Zhang et al. Large language models as data preprocessors. TaDA 2024.

Table Preprocessing

# ☞ Task Variety: Chain-of-Thought

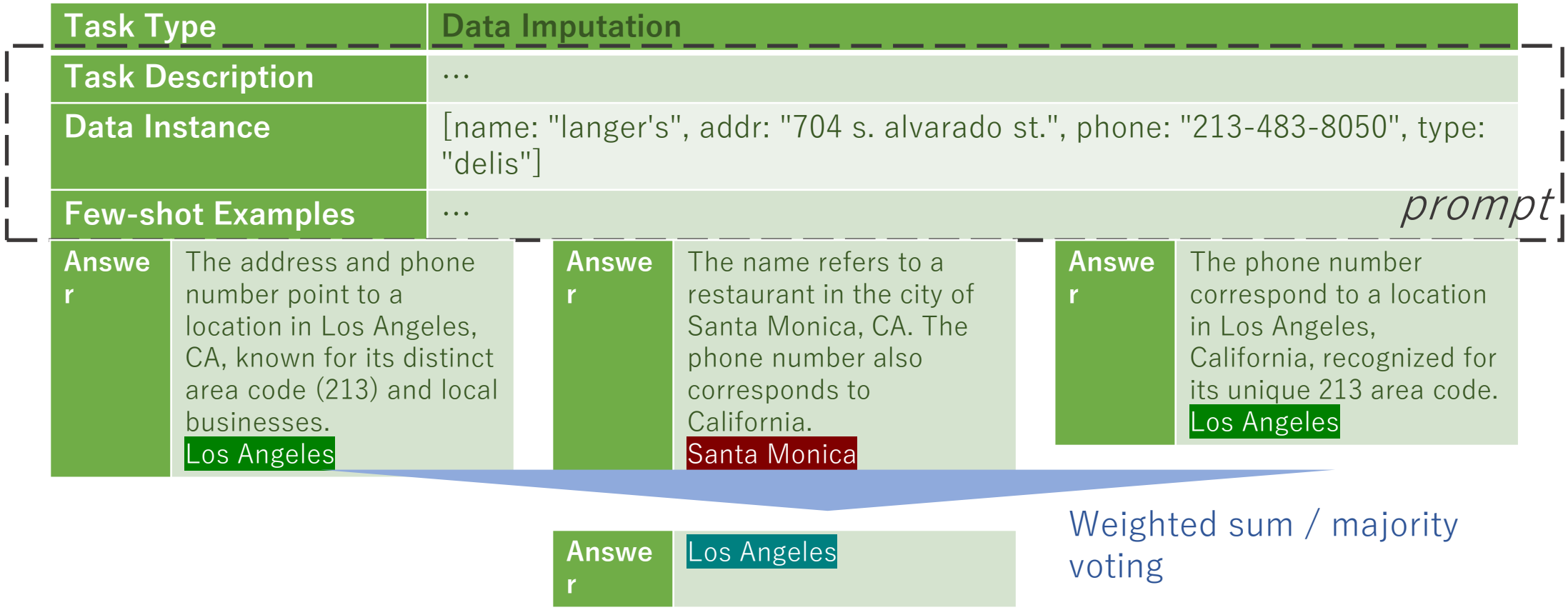
| Task Type | Data Imputation |
|---|---|
| **Task Description** | …<br>MUST answer each question in two lines. In the first line, you **give the reason** for the inference. In the second line, you ONLY **give the value** of the "city" attribute. |
| **Data Instance** | [name: "langer's", addr: "704 s. alvarado st.", phone: "213-483-8050", type: "delis"] |
| **Few-shot Examples** | Some examples are given below.<br>```<br>User:<br>Question 1: Record is [name: "carey's corner", addr: "1215 powers ferry rd.", phone: "770-933-0909", type: "hamburgers"]. What is the city?<br>Assistant:<br>Answer 1: The phone area code 770 corresponds to the Atlanta metropolitan area in Georgia, and "1215 Powers Ferry Rd." is an address located in Marietta, Georgia; therefore, the city is Marietta.<br>Marietta<br>…<br>*prompt* |
| **Answer** | The address and phone number point to a location in Los Angeles, CA, known for its distinct area code (213) and local businesses.<br>Los Angeles |

Zhang et al. Large language models as data preprocessors. TaDA 2024.

Table Preprocessing

# ☞ Task Variety:
# Chain-of-Thought (+Self-Consistency Decoding)

- Generate multiple reasoning answers and aggregate them.




**Chen. Large language models are few (1)-shot table reasoners. EACL 2023.**

Chen et al. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. TMLR 2023.

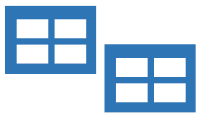
TA → Table QA

# ☞ Task Complexity

Sources of complexity

-  Task
-  Table

Strategies

-  Task decomposition
-  Table decomposition

# ☞ Task Complexity: Task Decomposition

| ? | ? | ? | ? |
|---|---|---|---|
| Friends Pizza | 2525 | Cash Visa MasterCard | 7:30AM |

Before decomposition

| Task Type | Column Type Annotation |
|---|---|
| Task Description | Classify the columns of a given table with only one of the following classes that are separated with comma:<br>description of event, description of restaurant, postal code, region of address … |
| Data Instance | Column 1 \|\| Column 2 \|\| Column 3 \|\| Column 4 ¥n<br>Friends Pizza \|\| 2525 \|\| Cash Visa MasterCard \|\| 7:30AM ¥n |
| Answer | name, number, payment, time |

**Korini and Bizer. Column type annotation using ChatGPT. TaDA 2023.** TU → Table Interpretation

Zhao et al. Large language models are complex table parsers. EMNLP 2023. TA → Table QA

Wang et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. ICLR 2024. TP → Table Transformation, TA → Table QA

Dong et al. OpenTE: Open-structure table extraction from text. ICCASP 2024. TU → Table Interpretation

# ☞ Task Complexity: Task Decomposition

**Sub-Task 1**

After decomposition

**Sub-Task 2**

| Sub-Task Type | Table Classification |
|---|---|
| Task Description | Your task is to **classify if a table** describes Restaurants, Events, Music Recordings, or Hotels. |
| Data Instance | Column 1 \|\| Column 2 \|\| Column 3 \|\| Column 4 ¥n<br>Friends Pizza \|\| 2525 \|\| Cash Visa MasterCard \|\| 7:30AM ¥n |
| Answer | Restaurant |

| Sub-Task Type | Column Classification |
|---|---|
| Task Description | Your task is to **classify the columns** of a given table with only one of the following classes that are separated with comma: name of restaurant, description of restaurant … |
| Data Instance | Column 1 \|\| Column 2 \|\| Column 3 \|\| Column 4 ¥n<br>Friends Pizza \|\| 2525 \|\| Cash Visa |
| Answer | name of restaurant, postal code, payment accepted, time |

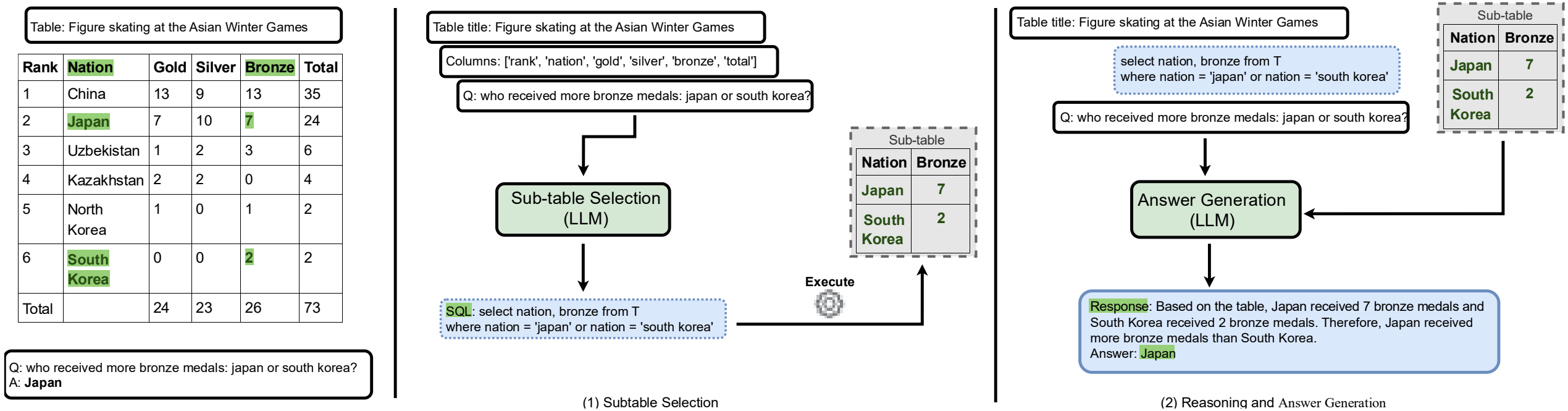
**Korini and Bizer. Column type annotation using ChatGPT. TaDA 2023.** — TU → Table Interpretation

Zhao et al. Large language models are complex table parsers. EMNLP 2023. — TA → Table QA

Wang et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. ICLR 2024. — TP → Table Transformation; TA → Table QA

Dong et al. OpenTE: Open-structure table extraction from text. ICCASP 2024. — TU → Table Interpretation

# ☞ Task Complexity: Table Decomposition




**Nahid and Rafiei. TabSQLify: Enhancing reasoning capabilities of LLMs through table decomposition. NAACL 2024.** TA → Table QA | TA → Text-to-SQL

Patnaik et al. Cabinet: Content relevance based noise reduction for table question answering. ICLR 2024. TA → Table QA

Jiang et al. StructGPT: A general framework for large language model to reason over structured data. EMNLP 2023. TA → Table QA | TA → Text-to-SQL

# ☞ Task Complexity:
# Table Decomposition (Progressive Prompting)

| Question | Report the number of wins in Grand Slam tournaments. | | |
|---|---|---|---|
| tournament | … | attn | career_w/l |
| Australian Open | … | 18 | 22-18 |
| Roland Garros | … | 14 | 11-14 |
| Wimbledon | … | 18 | 13-18 |
| US Open | … | 13 | 16-13 |
| Indian Wells | … | 15 | 20-15 |
| … | … | … | … |

Text transformation

| tournament | … | attn | career_w/l |
|---|---|---|---|
| Australian Open | … | 18 | 22-18 |
| Roland Garros | … | 14 | 11-14 |
| Wimbledon | … | 18 | 13-18 |
| US Open | … | 13 | 16-13 |
| Indian Wells | … | 15 | 20-15 |
| … | … | … | … |

| n_win |
|---|
| 22 |
| 11 |
| 13 |
| 16 |
| |
| |

| | | |
|---|---|---|
| Focus on **column selection**. | Incorporate both **column and row selection**. | Apply **additional operations** (e.g., aggregation functions and text operations). |
| Set the groundwork for understanding how to fetch specific data from a database. | Extract particular columns and filtering rows based on specified criteria, enhancing precision in data gathering. | Aggregation functions empower data summarization.<br>Text operations facilitate the manipulation and transformation of string data. |

Kong et al. OpenTab: Advancing large language models as open-domain table reasoners. ICLR 2024.

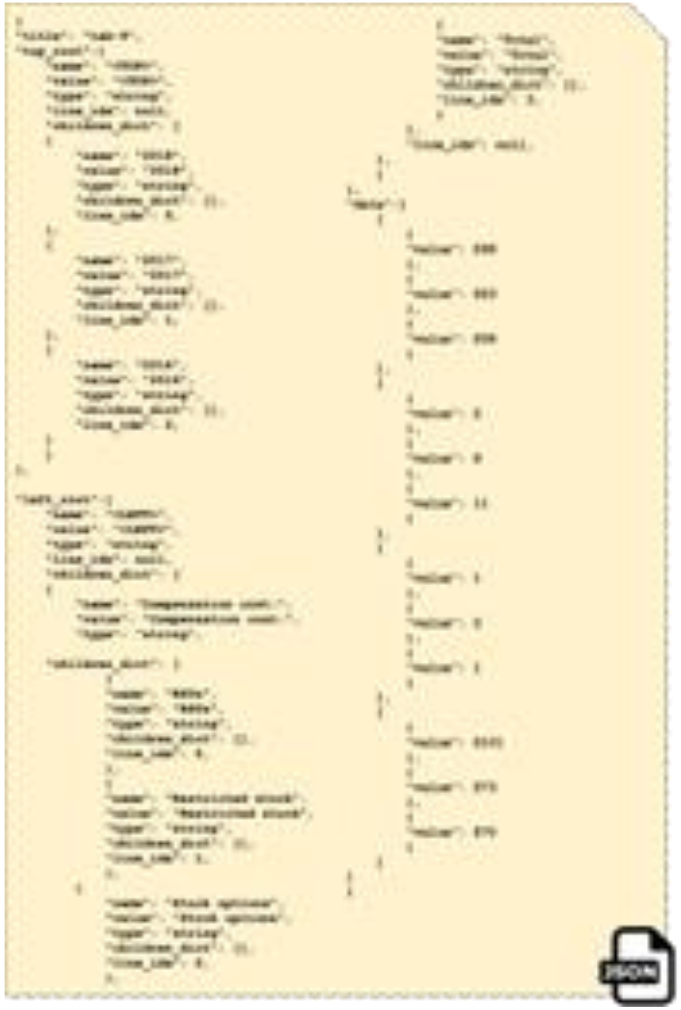
TA → Text-to-SQL

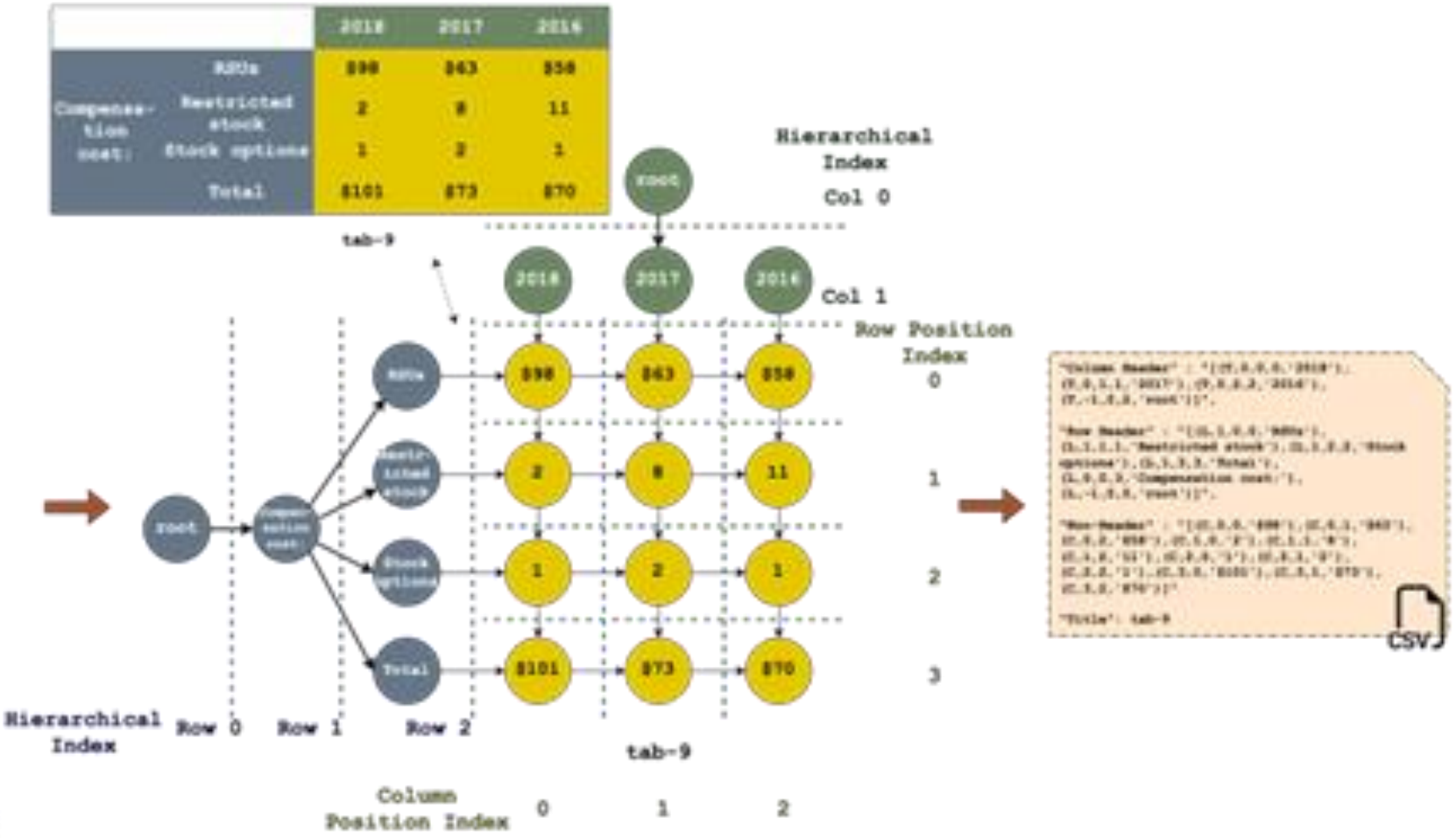
# ☞ Data Format: Table Encoding

| text (serialized) | spreadsheet | markup | key-value | program | image | embedded |
|---|---|---|---|---|---|---|
|  |   |    |  |  |   |   |

easy hard

# ☞ Data Format: Table Reconstruction






Zhao et al. Large language models are complex table parsers. EMNLP 2023.

TA → Table QA

# ☞ Data Volume: Instance Prompting

| name | city | addr | phone | Type |
|---|---|---|---|---|
| Langer's | ? | 704 S. Alvarado St. | 213-483-8050 | delis |
| Valetino | ? | 3115 Pico Blvd. | 310-829-4313 | Italian |
| Cafe Bizou | ? | 14016 Ventura Blvd. | 818/788-3536 | French |

| Task Type | Data Imputation | Task Type | Data Imputation | Task Type | Data Imputation |
|---|---|---|---|---|---|
| Task Description | You are a database engineer.<br><br>You are requested to infer the value of the "city" attribute based on the values of other attributes. | Task Description | You are a database engineer.<br><br>You are requested to infer the value of the "city" attribute based on the values of other attributes. | Task Description | You are a database engineer.<br><br>You are requested to infer the value of the "city" attribute based on the values of other attributes. |
| Data Instance | [name: "langer's", addr: "704 s. alvarado st.", phone: "213-483-8050", type: "delis"] *prompt* | Data Instance | [name: "valentino", addr: "3115 pico blvd.", phone: "310-829-4313", type: "italian"] *prompt* | Data Instance | [name: "cafe bizou", addr: "14016 ventura blvd.", phone: "818/788-3536", type: "french"] *prompt* |

Table Preprocessing

Zhang et al. Large language models as data preprocessors. TaDA 2024.

# ☞ Data Volume: Batch Prompting

| name | city | addr | phone | Type |
|---|---|---|---|---|
| Langer's | ? | 704 S. Alvarado St. | 213-483-8050 | delis |
| Valetino | ? | 3115 Pico Blvd. | 310-829-4313 | Italian |
| Cafe Bizou | ? | 14016 Ventura Blvd. | 818/788-3536 | French |

| Task Type | Data Imputation |
|---|---|
| Task Description | You are a database engineer.<br><br>You are requested to infer the value of the "city" attribute based on the values of other attributes. |
| Data Instance | [name: "langer's", addr: "704 s. alvarado st.", phone: "213-483-8050", type: "delis"]<br>[name: "valentino", addr: "3115 pico blvd.", phone: "310-829-4313", type: "italian"]<br>[name: "cafe bizou", addr: "14016 ventura blvd.", phone: "818/788-3536", type: "french"] |

*prompt*

**Standard Prompting**

```
# K-shot in-context exemplars
Q: {question}
A: {answer}

Q: {question}
A: {answer}

...
# One sample to inference
Q: Ali had $21. Leila gave him half of her
$  100. How much does Ali have now?
-----------------------------------------------
# Response
A: Leila gave 100/2=50 to Ali. Ali now has
   $21+$50 = $71. The answer is 71.
```

**Batch Prompting**

```
# K-shot in-context exemplars in K/b batches
Q[1]: {question}
Q[2]: {question}
A[1]: {answer}
A[2]: {answer}
```

*b*(=2) samples in one batch

```
...
# b samples in a batch to inference
Q[1]: Ali had $21. Leila gave him half of her
      $100. How much does Ali have now?
Q[2]: A robe takes 2 bolts of blue fiber and
      half that white fiber. How many bolts?
-----------------------------------------------
# Responses to a batch
A[1]: Leila gave 100/2=50 to Ali. Ali now has
      $21+$50 = $71. The answer is 71.
A[2]: It takes 2/2=1 bolt of white fiber. The
      total amount is 2+1=3. The answer is 3.
```

source: Cheng et al. Batch prompting: Efficient inference with large language model APIs. EMNLP 2023.

Zhang et al. Large language models as data preprocessors. TaDA 2024.

Table Preprocessing

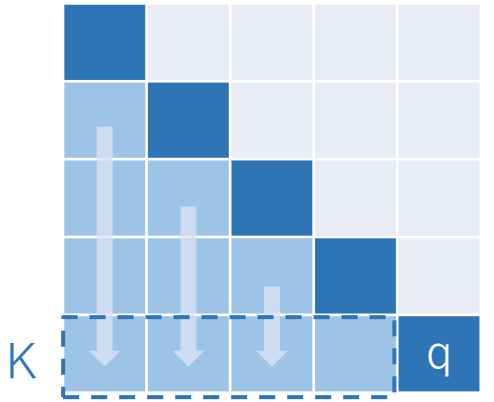
# ☞ Data Volume: Prefix Caching (+Instance Prompting)

- Use Automatic Prefix Caching (APC) in the vLLM library.

| | | | | | |
|---|---|---|---|---|---|
| **Task Description** | **You are a database engineer.**<br><br>**You are requested to** infer the value of the "city" attribute **based on the values of other attributes.** | **Task Description** | **You are a database engineer.**<br><br>**You are requested to** infer the value of the "city" attribute **based on the values of other attributes.** | **Task Description** | **You are a database engineer.**<br><br>**You are requested to** infer the value of the "city" attribute **based on the values of other attributes.** |
| **Data Instance** | [name: "langer's", addr: "704 s. alvarado st.", phone: "213-483-8050", type: "delis"] | **Data Instance** | [name: "valentino", addr: "3115 pico blvd.", phone: "310-829-4313", type: "italian"] | **Data Instance** | [name: "cafe bizou", addr: "14016 ventura blvd.", phone: "818/788-3536", type: "french"] |

*prompt* *prompt* *prompt*

Share prefix → Share KV cache




KV cache:
cache key vectors to speed up QKV attention in Transformers

q: query
K: key(s)

# Summary

- Prompt engineering is effective in improving the performance of table tasks.
- Zero-shot, few-shot, and chain-of-thought are useful across a variety of tasks.
- Task/table decomposition is recommended when dealing with complex tasks.
- Tables can be encoded in various formats. Reconstruction may help reduce difficulty.
- To speed up local deployment, users are suggested to use APC in the vLLM library.

# Q&A

\Orchestrating a brighter world

NEC

- On the Use of Large Language Models for Table Tasks
  - Prompting

大阪大学
OSAKA UNIVERSITY

Orchestrating a brighter world

NEC

大阪大学
OSAKA UNIVERSITY

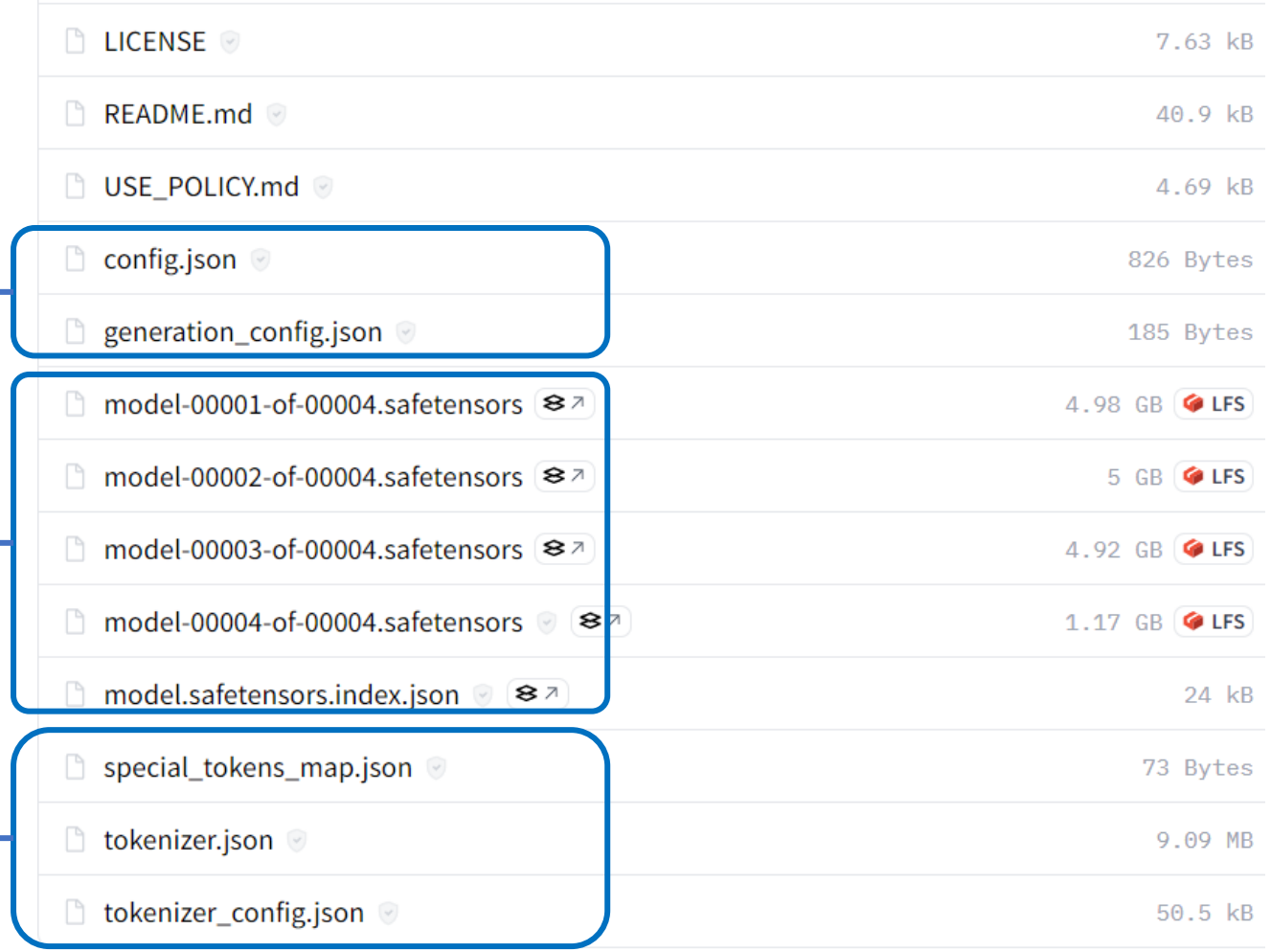
# On the Use of Large Language Models for Table Tasks

## - Fine-Tuning

# A Brief Tour of LLM

# Anatomy of an Open-Source LLM

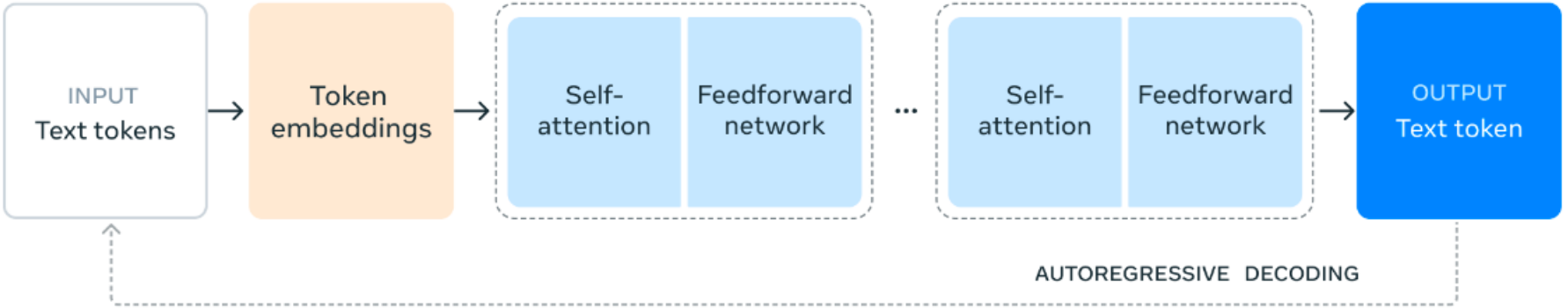
- Configuration Files
  - Model architecture hyperparameters
  - Default settings for text generation
- Model Weight Files
  - Model parameters
  - Index mapping model components to files
- Tokenizer Files:
  - Special token maps (e.g. start of text indicator)
  - Vocabulary File
  - Tokenizer settings



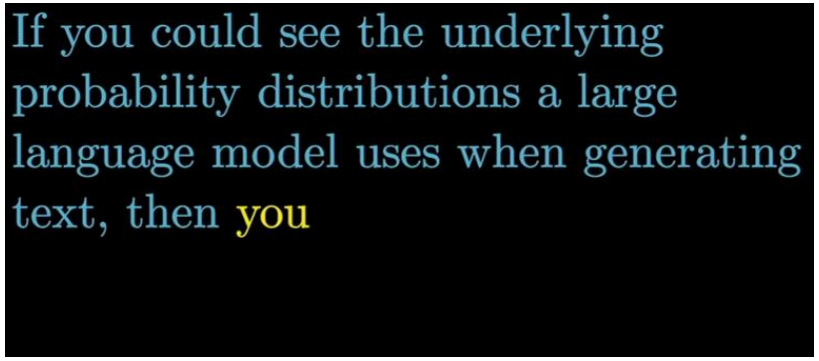

1. Composition of Llama 3.1 8B Model

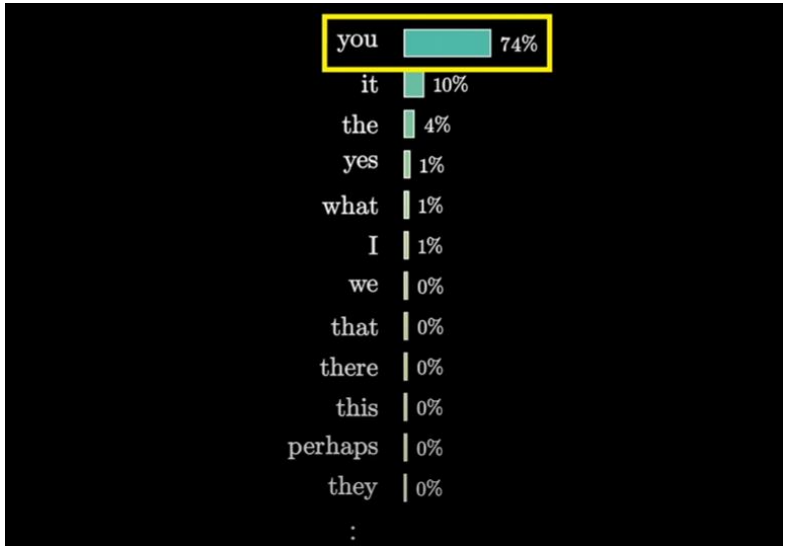
1: Meta: https://huggingface.co/meta-llama/Llama-3.1-8B/tree/main

# How LLMs Work




1. Typical Generation Process of Decoder-Only Transformer Models (Llama 3.1)

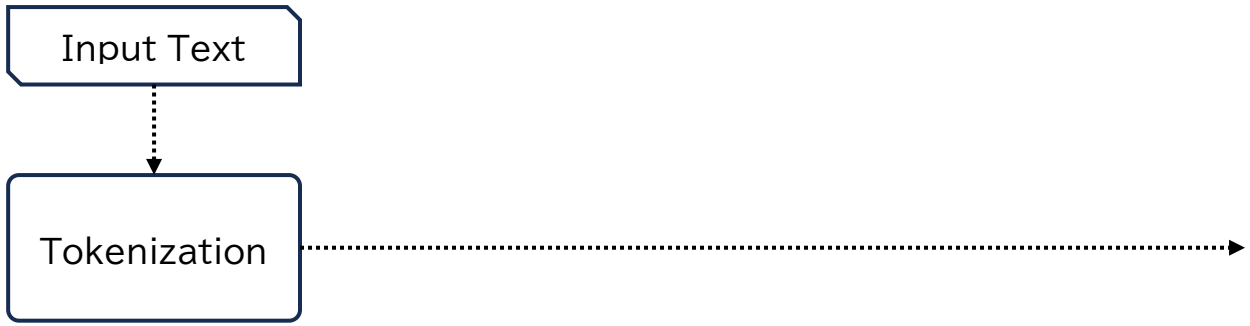


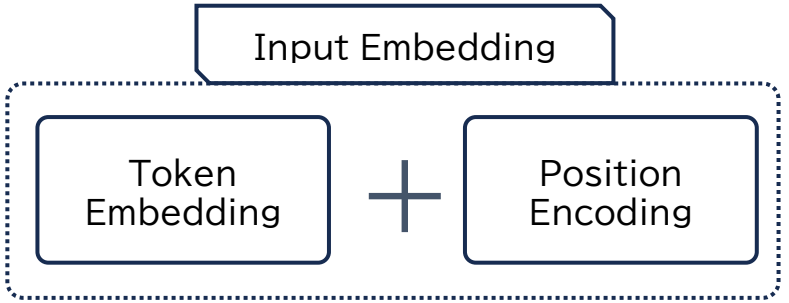




2. High-Level Visualization of Autoregressive Decoding

1: Meta https://ai.meta.com/blog/meta-llama-3-1/
2: 3Blue1Brown
https://www.3blue1brown.com/topics/neural-networks

# Dive into Forward Pass




LLMs compute the logits for the next token based on input text<newline>

[7454, 25153, 23864, 290, 148063, 395, 290, 2613, 6602, 4122, 402, 3422, 2201, 198]

1. GPT-4o Tokenization Demonstration

## Embedding Matrix: $W_E$

$d_{model}$
$N_{vocab}$

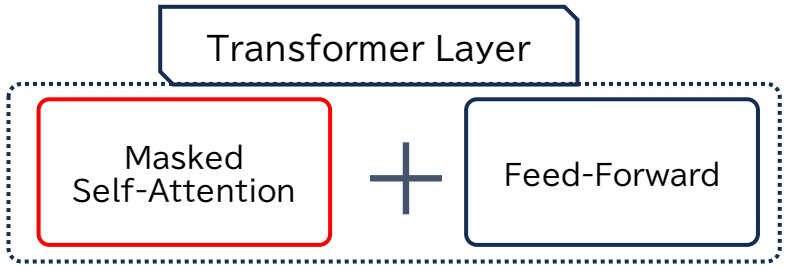
```
"hidden_size": 4096,
"vocab_size": 128256
"weight_map": {
  "lm_head.weight": "model-00004-of-00004.safetensors",
  "model.embed_tokens.weight": "model-00001-of-00004.safetensors",
```

## Rotary Position Embeddings (RoPE)

```
"rope_scaling": {
  "factor": 8.0,
  "low_freq_factor": 1.0,
  "high_freq_factor": 4.0,
  "original_max_position_embeddings": 8192,
  "rope_type": "llama3"
},
"rope_theta": 500000.0,
```

1. OpenAI: https://platform.openai.com/tokenizer

# Dive into Forward Pass




- Self-Attention

$$Q = XW_Q, \qquad K = XW_K, \qquad V = XW_V$$

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Masked Self-Attention

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V$$

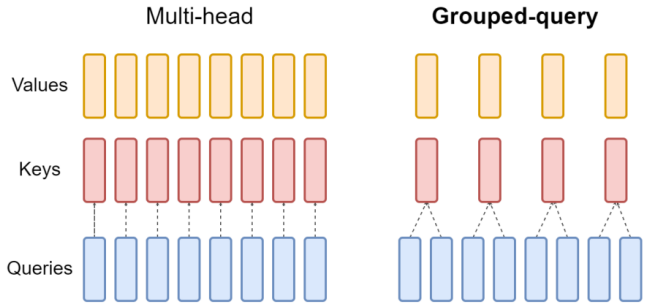
$$M_{i,j} = \begin{cases} 0, & j \le i \\ -\infty, & j > i \end{cases}$$

- Multi-Head Attention

$$Output = Concat(head_i)\, W_O$$

  - Each attention head computes a portion
  - The outputs are concatenated

- Grouped-Query Attention




1. Overview of grouped-query method

$W_K$ `"model.layers.0.self_attn.k_proj.weight"`
$W_O$ `"model.layers.0.self_attn.o_proj.weight"`
$W_Q$ `"model.layers.0.self_attn.q_proj.weight"`
$W_V$ `"model.layers.0.self_attn.v_proj.weight"`

```
"num_attention_heads": 32,
"num_hidden_layers": 32,
"num_key_value_heads": 8,
```

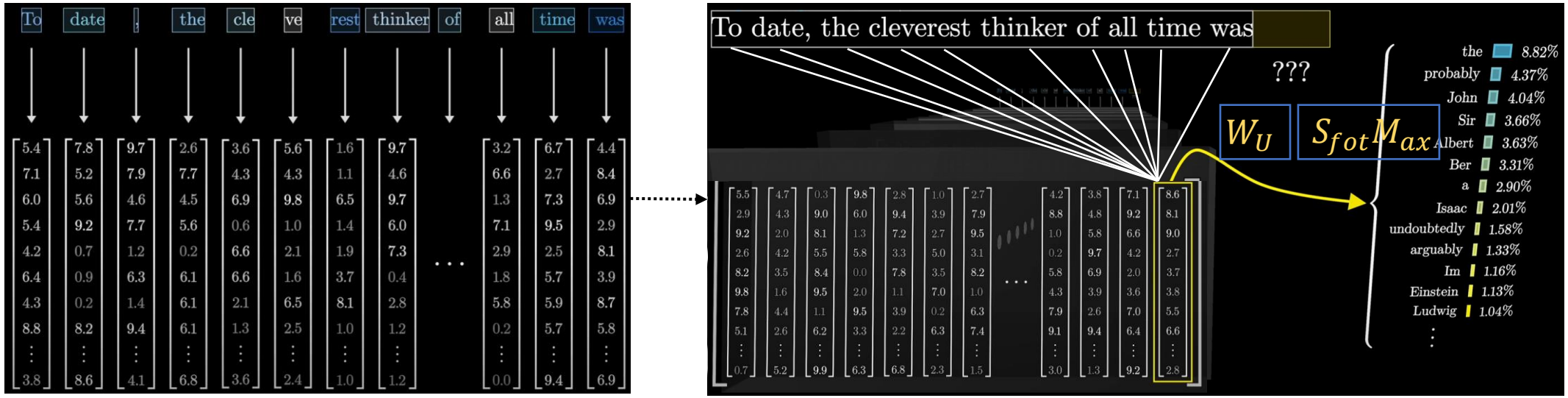
1: Ainslie et al. GQA: Training Generalized Mult Transformer Models from Multi-Head Checkpoints. EMNLP 2023

# Dive into Forward Pass

Output Layer/ Decoding

- Unembedding Matrix: $W_U$

$W_U$ "lm_head.weight": "model-00004-of-00004.safetensors"




1. Comparison between Input Embeddings and Final Outputs

- Tied/Untied Input-output Embeddings
  - If the unembedding matrix is the transformation of the embedding matrix or a separate matrix

```
"tie_word_embeddings": false,
```

1: 3Blue1Brown
https://www.3blue1brown.com/topics/neural-networks

# Dive into Forward Pass

Output Layer/ Decoding

```
"do_sample": true,
"temperature": 0.6,
"top_p": 0.9,
```

1. Default Decoding Settings for Llama3.1 8B

- Greedy Decoding
  Always selecting the option with the highest probability

- Temperature
  Controls the **sharpness** or **smoothness** of the distribution

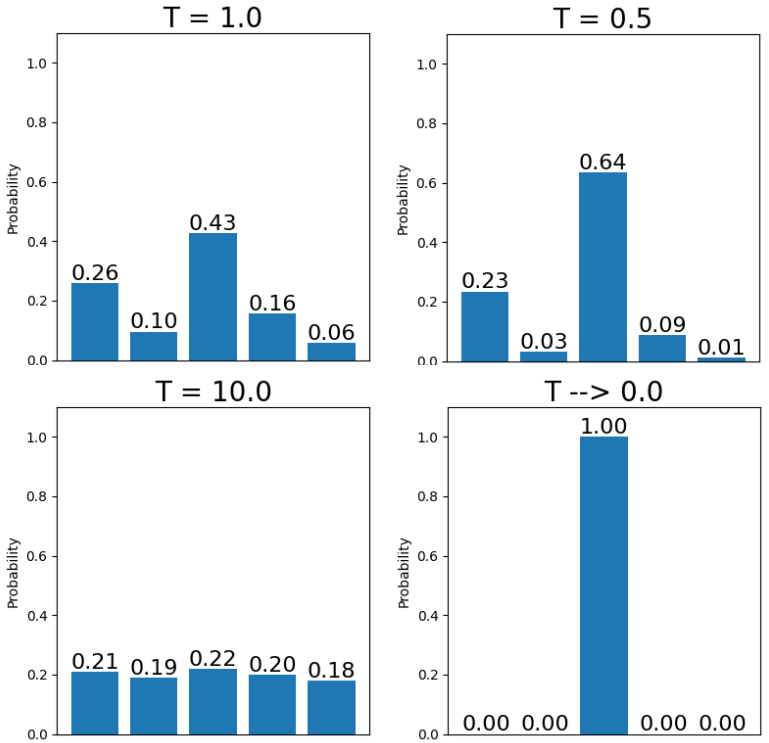
$$P_i = \frac{\exp(logit_i/T)}{\sum_j \exp(logit_j/T)}$$

  In practice, greedy decoding is used when T= 0.

- Top-p Sampling (Nucleus Sampling)
  - Cumulative Probability
  - Nucleus (Token Set) at Threshold p
  - Re-Normalization within the Nucleus

- Top-k Sampling, Beam Search, etc.

$Logtis: [1.5, 0.5, 2.0, 1.0, 0.0]$




2. Demonstration of SoftMax with Temperature

1: Meta: https://huggingface.co/meta-llama/Llama-3.1-8B/tree/main

# Recap on Llama3.1 8B

```
"attention_bias": false,
"attention_dropout": 0.0,
"bos_token_id": 128000,
"eos_token_id": 128001,
"hidden_act": "silu",
"hidden_size": 4096,
"initializer_range": 0.02,
"intermediate_size": 14336,
"max_position_embeddings": 131072,
"mlp_bias": false,
"model_type": "llama",
"num_attention_heads": 32,
"num_hidden_layers": 32,
"num_key_value_heads": 8,
"pretraining_tp": 1,
"rms_norm_eps": 1e-05,
"model.layers.0.mlp.down_proj.weight"
"model.layers.0.mlp.gate_proj.weight"
"model.layers.0.mlp.up_proj.weight":
"torch_dtype": "bfloat16",
"metadata": {
  "total_size": 16060522496
},
```

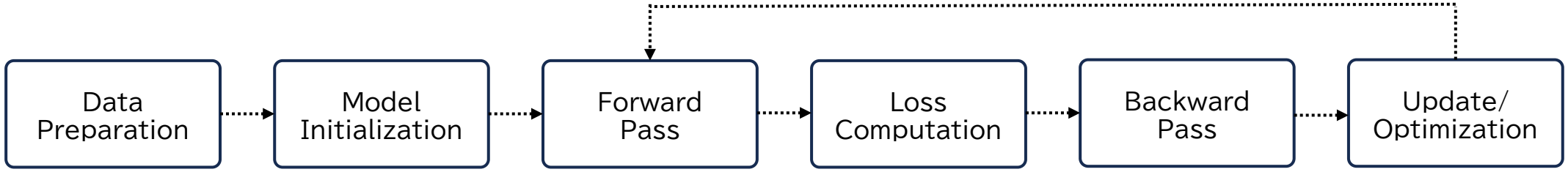
$d_{model}$ $d_{FF}$ $d_k = d_{model}/N_{heads}$ *Context Window* $N_{heads}$ $N_{att.layers}$ $N_{kv\_heads}$ $W_{Up}$ $W_{Gate}$ $W_{Down}$ $In\ Bytes = 8{,}030{,}261{,}248 * 2$

1. Llama 3.1-8B Configuration Breakdown

| | | | | Per Layer | Layers | |
|---|---|---|---|---|---|---|
| | | Embedding Matrix | $W_E$ | $N_{vocab} * d_{model} = 128{,}256 * 4{,}096 = 525{,}336{,}576$ | 1 | 525,336,576 |
| Transformer Layer | Multi-head Attention | Query Matrix | $W_Q$ | $d_{model} * d_k * N_{heads} = 4{,}096 * 128 * 32 = 16{,}777{,}216$ | $N_{layers} = 32$ | 6,979,321,856 |
| | | Key Matrix | $W_K$ | $d_{model} * d_k * N_{kvheads} = 4{,}096 * 128 * 8 = 4{,}194{,}304$ | | |
| | | Value Matrix | $W_V$ | $d_{model} * d_k * N_{kvheads} = 4{,}096 * 128 * 8 = 4{,}194{,}304$ | | |
| | | Output Matrix | $W_O$ | $d_{model} * d_{model} = 4096 * 4{,}096 = 16{,}777{,}216$ | | |
| | MLP | Up Projection | $W_{Up}$ | $d_{model} * d_{FF} = 4096 * 14{,}336 = 58{,}720{,}256$ | | |
| | | Gate Projection | $W_{Gate}$ | $d_{model} * d_{FF} = 4096 * 14{,}336 = 58{,}720{,}256$ | | |
| | | Down Projection | $W_{Down}$ | $d_{FF} * d_{model} = 14{,}336 * 4096 = 58{,}720{,}256$ | | |
| | | Unembedding Matrix | $W_U$ | $d_{model} * N_{vocab} = 4{,}096 * 128{,}256 = 525{,}336{,}576$ | 1 | 525,336,576 |
| | | LayerNorm | $W_{LN}$ | $d_{model} = 4{,}096$ | $2 * N_{layers} + 1 = 65$ | 266,240 |
| | | | | | Total | 8,030,261,248 |

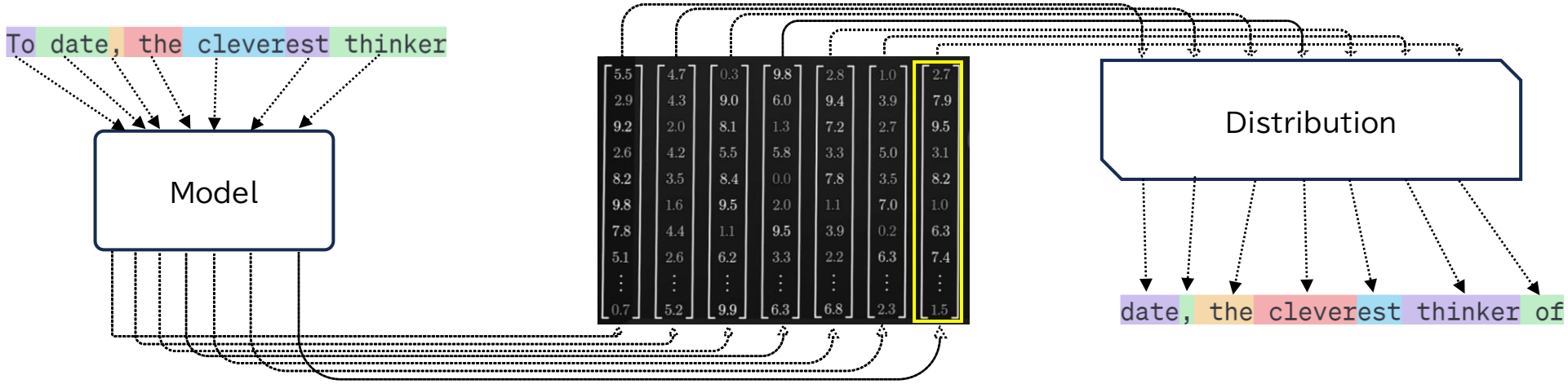
1: Meta https://huggingface.co/meta-llama/Llama-3.1-8B/tree/main

# Fundamentals of LLMs Training




- Teacher Forcing
  - Uses ground truth next tokens as inputs for each time step
- Target Shifting
  - Processes entire input sequence in one forward pass.

$$X = [x_1, x_2, \ldots, x_L], \qquad Y = [x_2, x_3, \ldots, x_{L+1}]$$



# Pre-Training and Fine-Tuning

- Pre-Training
  - Initialization
    Randomly initialize weights
  - Data
    Large scale, diverse text corpus
  - Purpose
    Learning general language representations and knowledge

The benefits of a healthy diet include

improved energy levels, better mood, and reduced risk of chronic diseases.

Llama3.1-8B

- Fine-Tuning
  - Initialization
    Load pre-trained weights
  - Data
    Task-specific or domain-specific
  - Purpose
    Following instructions or engaging in conversations

<|user|> List the benefits of a healthy diet.
<|endoftext|><|assistant|>

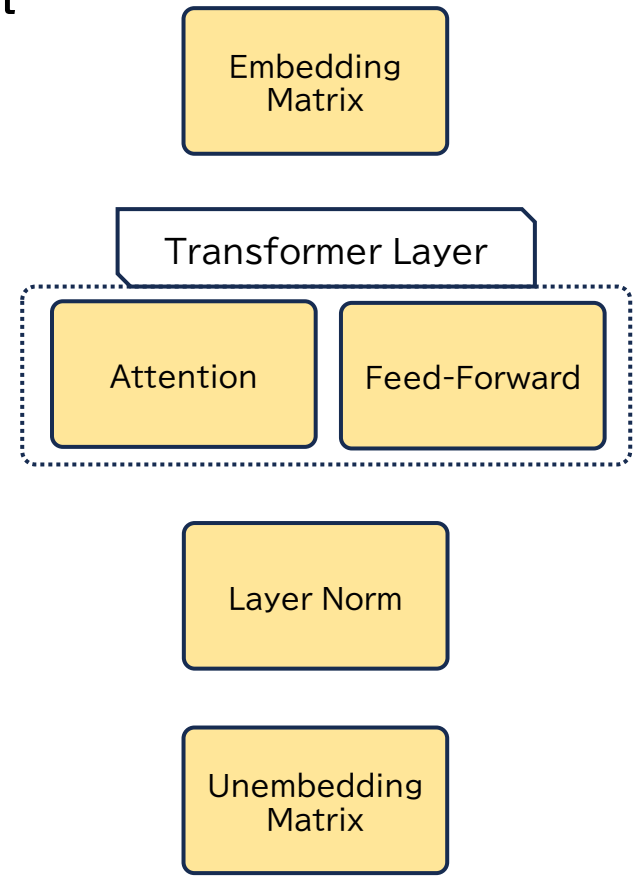
A healthy diet offers numerous benefits, including:
1. Improved energy levels
...

Llama-3.1-8B-Instruct

# Fine-Tuning Methods

# Full Parameter Fine-Tuning

- Adjusting **all the parameters** of a pre-trained model
- Advantages
  - Maximum Adaptability
  - Potential for Higher Performance
- Disadvantages
  - Computationally Intensive
  - Risk of Overfitting
- Use Case
  - Sufficient training data is available
  - The new task is significantly different

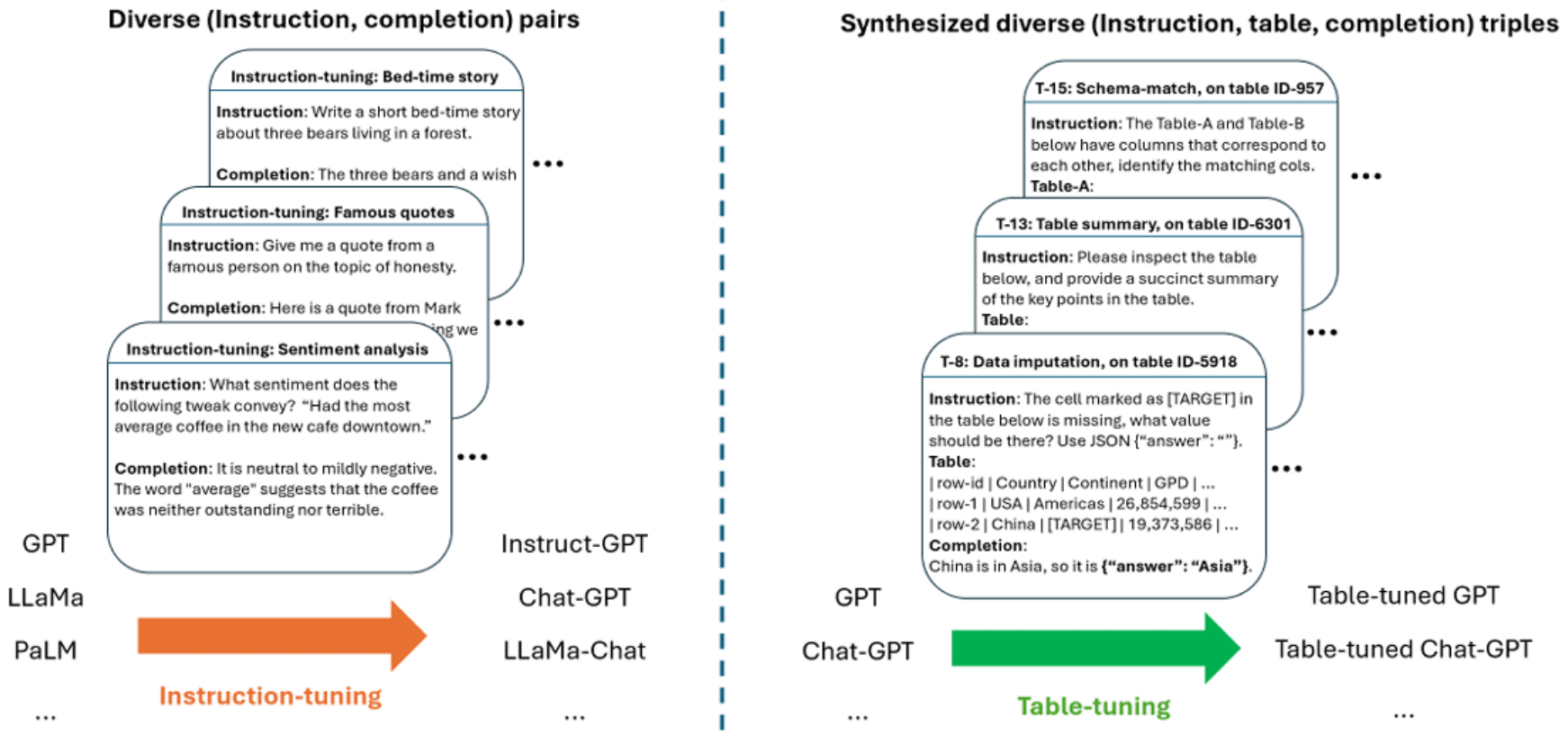


# Table-GPT

Table Preprocessing | Table Understanding | Table Analysis

GPT3.5 ChatGPT

- Extensive Task Coverage
- Synthesis-then-Augment
- Comprehensive Analysis




Li et al. Table-GPT: Table-tuned GPT for Diverse Table Tasks. SIGMOD 2024

# Table-GPT

## Test Only

| | |
|---|---|
| T-1: Missing-value identification (MV) | Identify the row and column position of the only missing cell in a given table |
| T-2: Column-finding (CF) | Identify the column-name of a specific value that appears only once in a given table |
| T-3: Table-QA (TQA) | Answer a natural-language question based on the content of a table ([11, 42, 49]) |
| T-4: Column type annotation (CTA) | Find the semantic type of a column, from a given list of choices ([16, 25, 63]) |

## Train / Test

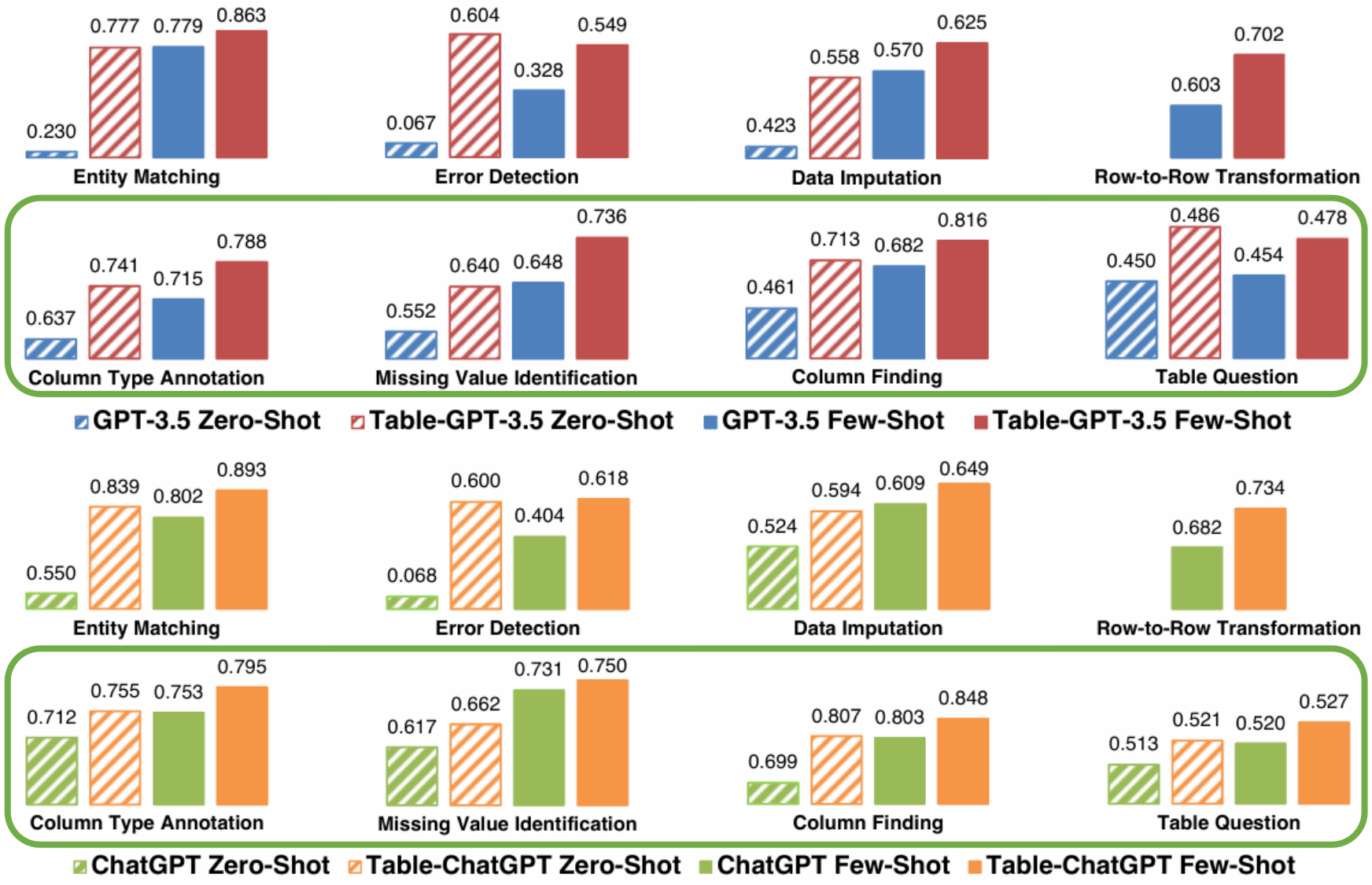
| | |
|---|---|
| T-5: Row-to-row transform (R2R) | Transform table data based on input/output examples ([23, 24, 27]) |
| T-6: Entity matching (EM) | Match rows from two tables that refer to the same real-world entity ([32, 38, 41, 66]) |
| T-7: Schema matching (SM) | Match columns from two tables that refer to the same meaning ([30, 36, 44]) |
| T-8: Data imputation (DI) | Predict the missing values in a cell based on the table context ([7, 37]) |
| T-9: Error detection (ED) | Detect data values in a table that is a likely error from misspelling ([14, 45]) |

## Train Only

| | |
|---|---|
| T-10: List extraction (LE) | Extract a structured table, from a list that lacks explicit column delimiters [9, 13, 19] |
| T-11: Head value matching (HVM) | Match column-headers with its data values drawn from the same table |
| T-12: Natural-language to SQL (NS) | Translate a natural-language question on a table into a SQL query ([62, 65]) |
| T-13: Table summarization (TS) | Produce a natural-language summary for the content in a table |
| T-14: Column augmentation (CA) | Augment a table with additional columns compatible with a given table |
| T-15: Row augmentation (RA) | Augment a table with additional rows compatible with a given table |
| T-16: Row/column swapping (RCSW) | Manipulate a given table, by swapping the position of two rows or columns |
| T-17: Row/column filtering (RCF) | Manipulate a given table, by filtering on given rows or columns |
| T-18: Row/column sorting (RCS) | Manipulate a given table, by performing sorting on given rows or columns |

Li et al. Table-GPT: Table-tuned GPT for Diverse Table Tasks. SIGMOD 2024

# Table-GPT




Li et al. Table-GPT: Table-tuned GPT for Diverse Table Tasks. SIGMOD 2024

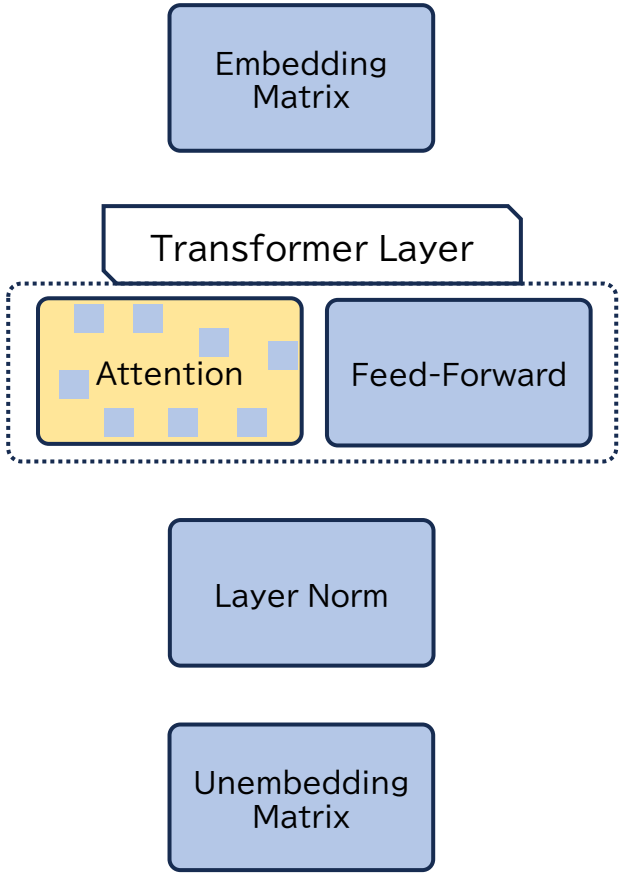
# Parameter-Efficient Fine-Tuning (PEFT)

- Adjust only a **small portion** of parameters
- Introduce additional lightweight modules

- Advantages
  - Resource Efficiency
  - Flexibility

- Disadvantages
  - Limited Capacity
  - Complex Implementation

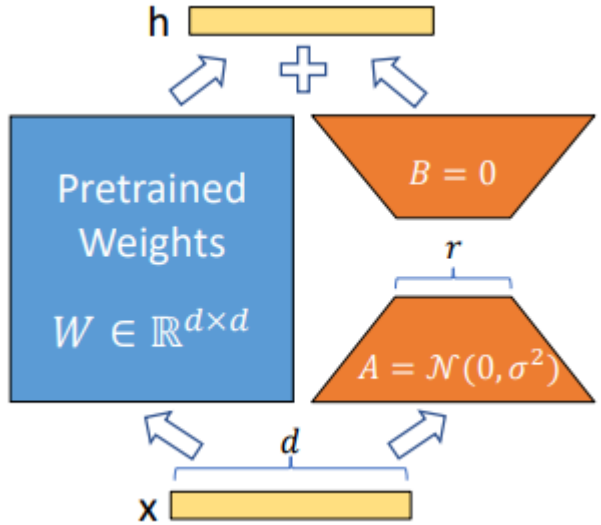
- Use Case
  - Task-specific data is limited
  - Computational resources are limited



# LoRA (Low-Rank Adaptation)

- Updates only low-rank matrices added to the model weights
- Advantages
  - Parameter Efficiency
  - Memory Efficiency
  - Modularity
  - Adaptability
- Disadvantages
  - Hyperparameter Optimization Needed

| | Weight Type | $r=1$ | $r=2$ | $r=4$ | $r=8$ | $r=64$ |
|---|---|---|---|---|---|---|
| WikiSQL(±0.5%) | $W_q$ | 68.8 | 69.6 | 70.5 | 70.4 | 70.0 |
| | $W_q, W_v$ | 73.4 | 73.3 | 73.7 | 73.8 | 73.5 |
| | $W_q, W_k, W_v, W_o$ | 74.1 | 73.7 | 74.0 | 74.0 | 73.9 |
| MultiNLI (±0.1%) | $W_q$ | 90.7 | 90.9 | 91.1 | 90.7 | 90.7 |
| | $W_q, W_v$ | 91.3 | 91.4 | 91.3 | 91.6 | 91.4 |
| | $W_q, W_k, W_v, W_o$ | 91.2 | 91.7 | 91.7 | 91.5 | 91.4 |




$$h = W_0 x + \Delta W x = W_0 x + BAx$$

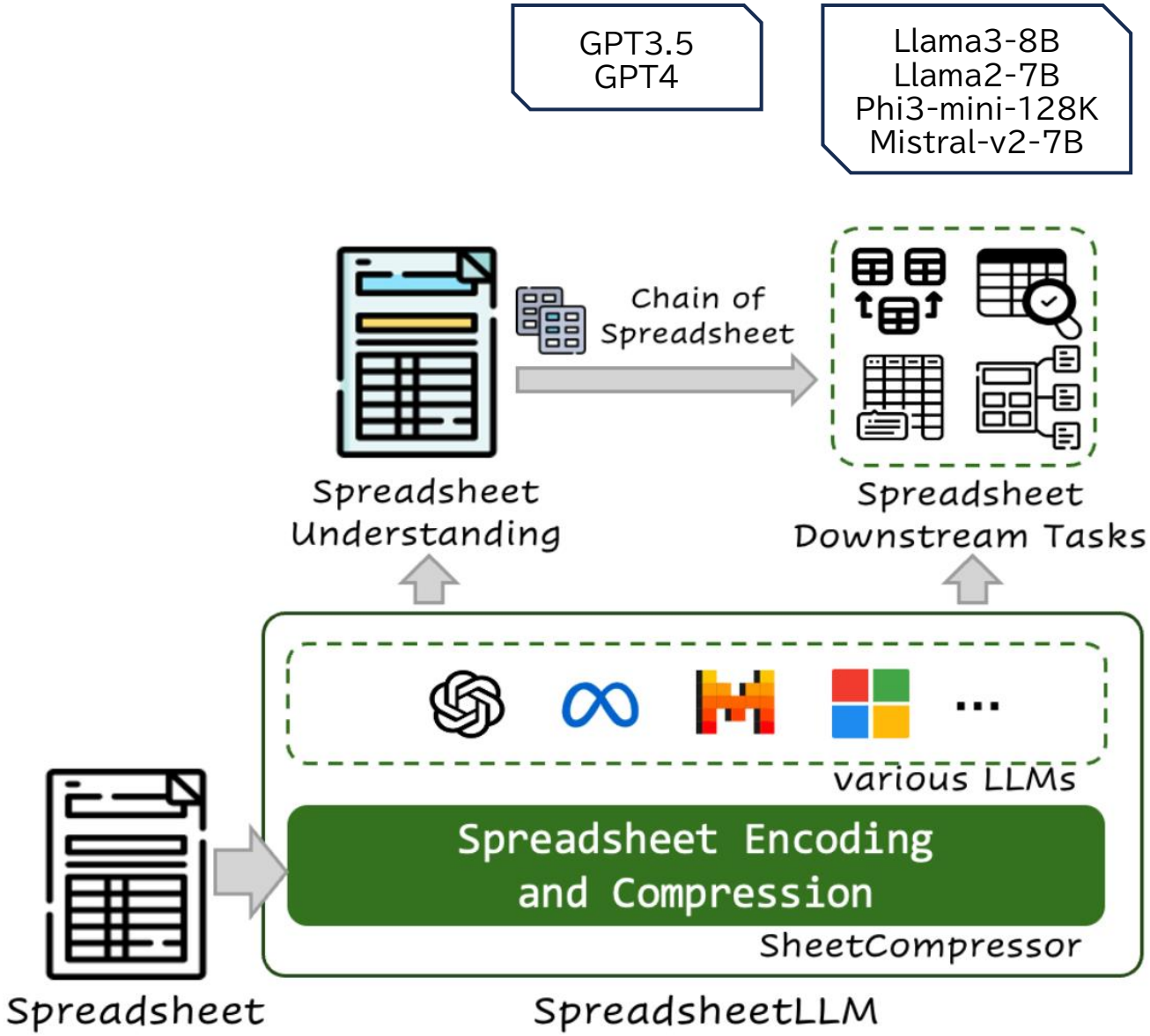
$$W = W_0 + BA$$

Hu et al. LoRA: Low-Rank Adaptation of Large Language Models. ICLR 2022

# SpreadSheetLLM

Table Understanding | Table Analysis

- Innovative Spreadsheet Encoding
- Chain of Spreadsheet Methodology
- Comprehensive Spreadsheet Analysis



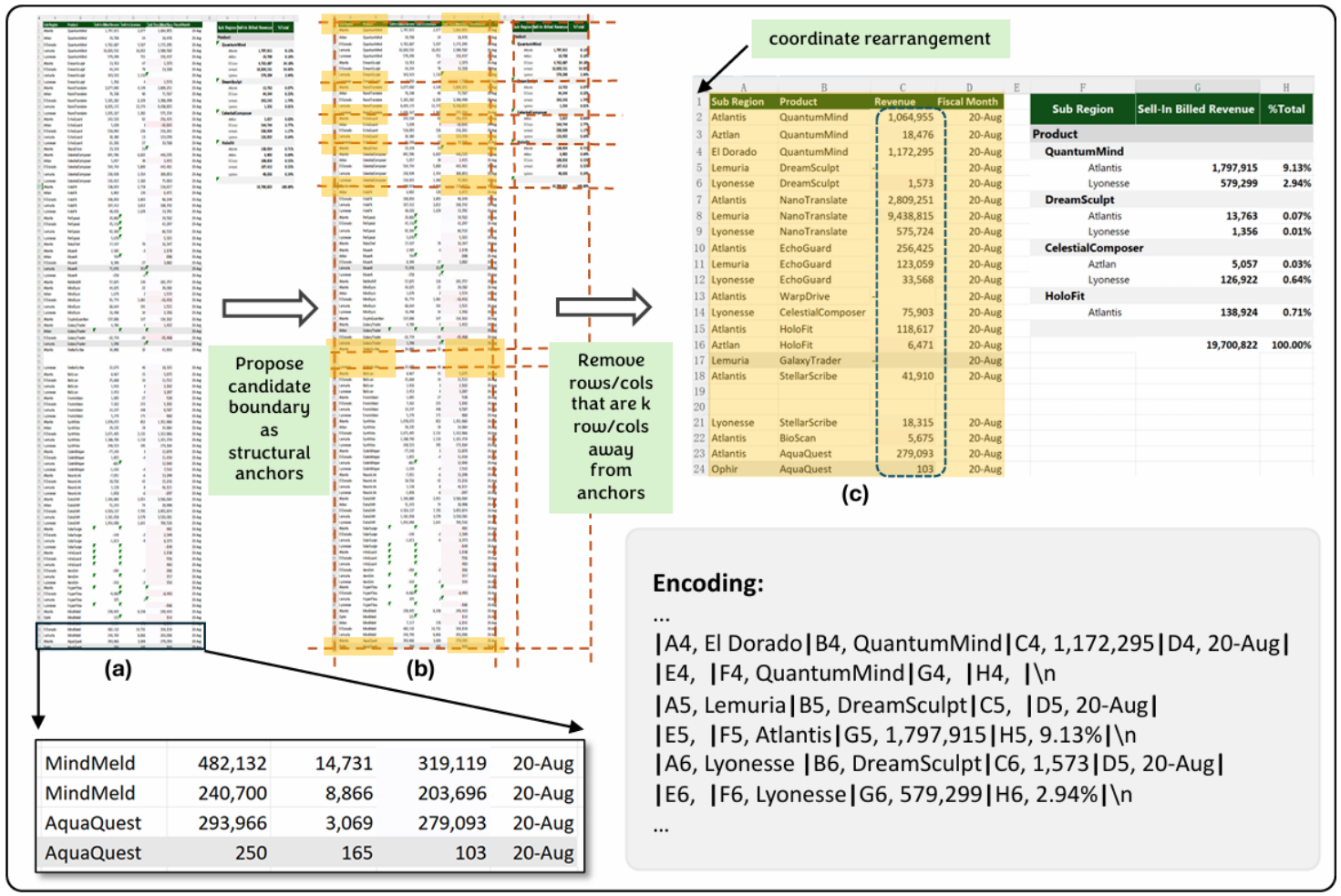

Tian et al. SpreadSheetLLM: Encoding Spreadsheets for Large Language Models. arXiv 2024

# SpreadSheetLLM

- Structural-Anchor-Based Extraction

  - Identifies key rows and columns at table boundaries

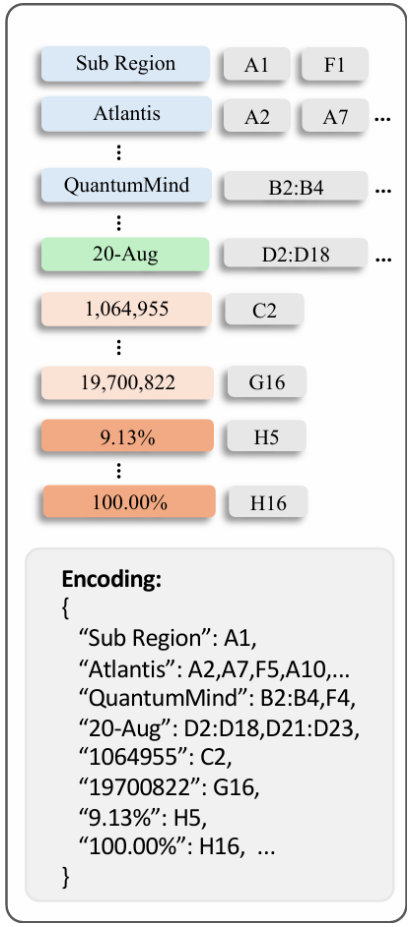
  - Removes distant rows & columns to create a “skeleton” of spreadsheets



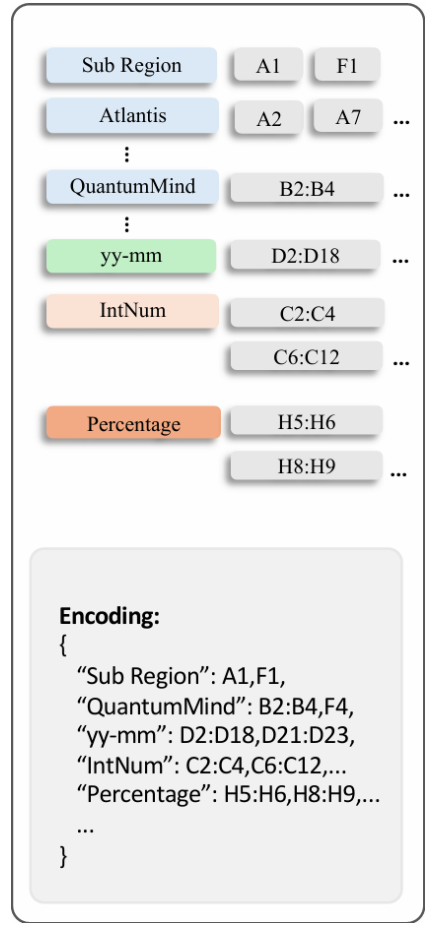

Tian et al. SpreadSheetLLM: Encoding Spreadsheets for Large Language Models. arXiv 2024

# SpreadSheetLLM

- Inverted-index Translation
  - Translates spreadsheet data into a dictionary-like format
  - Each unique value is stored once, with corresponding cell addresses indexed
  - Empty cells are excluded from the encoding




- Data-format-aware Aggregation
  - Groups cells into clusters
  - Recognizes and aggregates formats




Tian et al. SpreadSheetLLM: Encoding Spreadsheets for Large Language Models. arXiv 2024

# SpreadSheetLLM

- Results: Table Detection

| Model & Method | Small | Medium | Large | Huge | All |
|---|---|---|---|---|---|
| **ICL** | | | | | |
| Mistral-v2 | 0.071 | 0.013 | 0.029 | 0.017 | 0.036 |
| GPT4 | 0.318 | 0.292 | 0.090 | 0.000 | 0.154 |
| GPT4-compress | 0.480 | 0.454 | 0.373 | 0.330 | 0.410 |
| **Fine-tune** | | | | | |
| Llama3 | 0.715 | 0.765 | 0.290 | 0.000 | 0.471 |
| Llama2 | 0.557 | 0.378 | 0.107 | 0.000 | 0.280 |
| Phi3 | 0.604 | 0.481 | 0.201 | 0.130 | 0.330 |
| Mistral-v2 | 0.700 | 0.784 | 0.472 | 0.123 | 0.542 |
| GPT4 | 0.779 | 0.707 | 0.288 | 0.000 | 0.520 |
| Llama3-compress | 0.825 | 0.768 | 0.664 | 0.617 | 0.719 |
| Llama2-compress | 0.710 | 0.722 | 0.633 | 0.578 | 0.660 |
| Phi3-compress | 0.800 | 0.673 | 0.624 | 0.675 | 0.689 |
| Mistral-v2-compress | 0.778 | 0.729 | 0.686 | 0.744 | 0.726 |
| GPT3.5-compress | 0.795 | 0.649 | 0.600 | 0.680 | 0.680 |
| GPT4-compress | 0.810 | **0.832** | 0.718 | 0.690 | 0.759 |
| -w/o Aggregation | **0.864** | 0.816 | **0.739** | **0.753** | **0.789** |
| TableSense-CNN | 0.785 | 0.788 | 0.567 | 0.561 | 0.666 |

- Results: Spreadsheet QA

| Model | Accuracy |
|---|---|
| TAPEX | 0.378 |
| Binder | 0.622 |
| GPT4 | 0.466 |
| GPT4-compress-w/o splitting | 0.651 |
| GPT4-compress-w/o splitting-FT | 0.694 |
| GPT4-compress | 0.684 |
| GPT4-compress-FT | **0.743** |

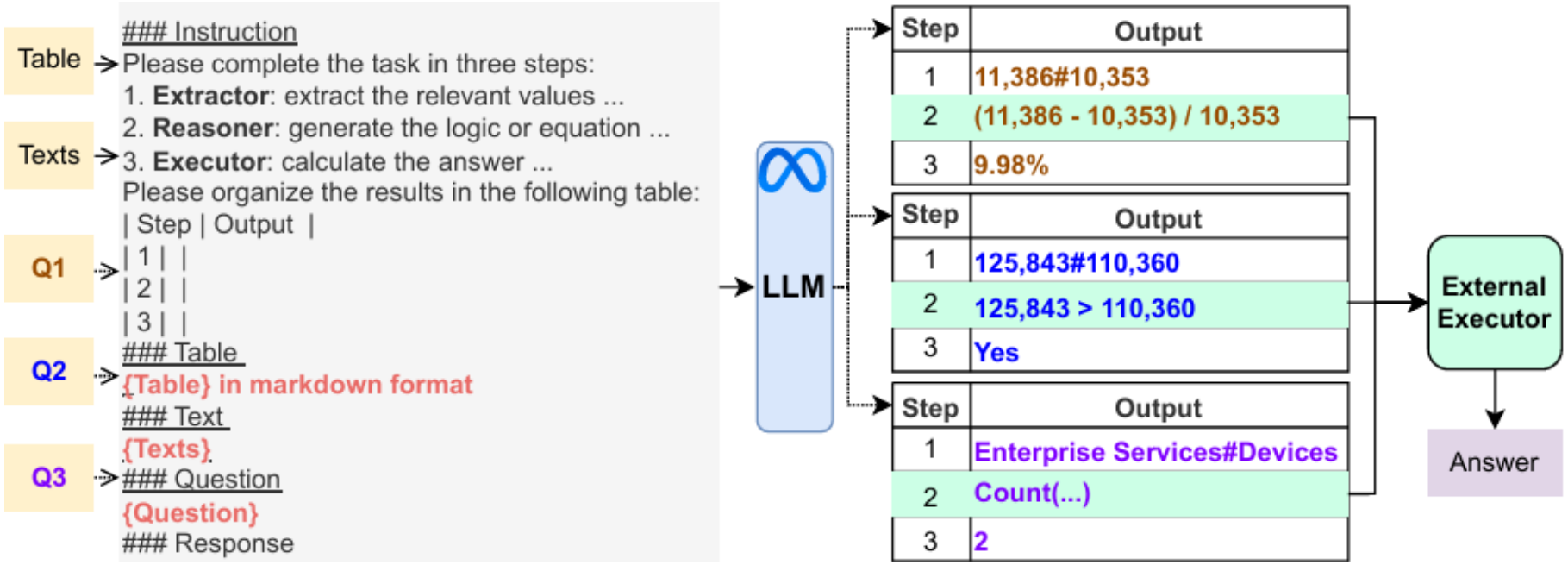
Tian et al. SpreadSheetLLM: Encoding Spreadsheets for Large Language Models. arXiv 2024

# TAT-LLM

Table Analysis

- Hybrid Tabular and Textual QA
- Step-wise Pipeline
- External Executor

Llama2-7B-Chat



# TAT-LLM

**Algorithm 1** : External Executor

**Input** $O_1$: the output of *Extractor*; $O_2$: the output of *Reasoner*; $O_3$: the output of *Executor*; $Q_t$: the predicted question type.

$answer \leftarrow O_3$
**if** $O_2$ is a valid arithmetic equation **then**
    $answer \leftarrow round(eval(O_2), 4)$
**else if** "#" in $O_2$ **then** # multiple values are separated with "#"
    $arr \leftarrow O_2.split("\#")$
    $answer \leftarrow len(arr)$
**else if** ">" in $O_2$ or "<" in $O_2$ **then**
    $answer \leftarrow eval(O_2)$
**else if** $O_2$ = "N.A." **then**
    **if** $Q_t$ = "Span" **then**
        $answer \leftarrow O_1$
    **else if** $Q_t$ = "Multiple Spans" **then**
        $arr \leftarrow O_1.split("\#")$ # spans are separated with "#"
        $answer \leftarrow arr$
    **end if**
**end if**

| Model | FinQA | TAT-QA | | TAT-DQA | |
|---|---|---|---|---|---|
| | EM | EM | $F_1$ | EM | $F_1$ |
| GPT-4 | 63.91 | 71.92 | 79.71 | 64.46 | 72.20 |
| TAT-LLM$_{All}$ (7B) | | | | | |
| w/o *External Executor* | 48.47 | 58.69 | 67.21 | 54.84 | 63.68 |
| w *External Executor* | 65.13 | 76.49 | 85.13 | 71.38 | 80.24 |
| gains (+) | 16.66 | 17.80 | 17.92 | 16.54 | 16.56 |
| TAT-LLM$_{All}$ (13B) | | | | | |
| w/o *External Executor* | 60.05 | 62.60 | 70.73 | 59.95 | 68.61 |
| w *External Executor* | 71.75 | 76.79 | 85.05 | 71.86 | 80.50 |
| gains (+) | 11.70 | 14.19 | 14.32) | 11.91 | 11.89 |
| TAT-LLM$_{All}$ (70B) | | | | | |
| w/o *External Executor* | 70.10 | 76.61 | 83.55 | 71.74 | 78.99 |
| w *External Executor* | **76.81** | **81.42** | **88.49** | **76.55** | **83.90** |
| gains (+) | 6.71 | 4.81 | 4.94 | 4.81 | 4.91 |

# TAT-LLM

| Type | Model | EM |
|---|---|---|
| **Human Expert Performance** | | 91.16 |
| **Fine-tuned** | Longformer | 21.90 |
| | NeRd | 48.57 |
| | FinQANet$_{BERT}$ | 50.00 |
| | DyRRen$_{BERT}$ | 59.37 |
| | FinQANet$_{RoBERTa}$ | 61.24 |
| | ELASTIC$_{RoBERTa}$ | 62.66 |
| | DyRRen$_{RoBERTa}$ | 63.30 |
| **Zero-shot** | Vicuna (7B) | 10.11 |
| | LLaMA 2-Chat (7B) | 15.43 |
| | LLaMA 2-Chat (70B) | 32.17 |
| | MAmmoTH (70B) | 36.09 |
| | WizardMath (70B) | 47.25 |
| | GPT3.5-Turbo | 58.00 |
| | GPT-4 | 63.91 |
| **Ours** | TAT-LLM (7B) | (+0.69) **64.60** |

Table 2: Performance of our TAT-LLM model and compared models on the test set of FinQA. Best results are marked in bold and numbers in red indicate the improvement over the underlined second-best results.

| Type | Model | EM | $F_1$ |
|---|---|---|---|
| **Human Expert Performance** | | 84.1 | 90.8 |
| **Fine-tuned** | TagOp | 50.10 | 58.00 |
| | TeaBReaC | 55.80 | 63.80 |
| | KIQA | 58.20 | 67.40 |
| | FinMath | 58.30 | 68.20 |
| | GANO | 61.90 | 72.10 |
| | MHST | 63.60 | 72.70 |
| | UniPCQA | 63.90 | 72.20 |
| | SoarGraph | 65.40 | 75.30 |
| | UniRPG | 67.20 | 76.00 |
| | RegHNT | 70.30 | 77.90 |
| | MVGE | 70.90 | 79.10 |
| **Zero-shot** | Vicuna (7B) | 32.53 | 40.97 |
| | LLaMA 2-Chat (7B) | 37.16 | 45.37 |
| | MAmmoTH (70B) | 38.97 | 46.51 |
| | WizardMath (70B) | 39.63 | 45.28 |
| | LLaMA 2-Chat (70B) | 45.94 | 53.80 |
| | GPT3.5-Turbo | 59.47 | 68.11 |
| | GPT-4 | 71.92 | 79.71 |
| **Ours** | TAT-LLM (7B) | (+2.64) **74.56** | (+3.17) **82.88** |

Table 3: Performance of our TAT-LLM model and compared models on the test set of TAT-QA.

| Type | Model | EM | $F_1$ |
|---|---|---|---|
| **Human Expert Performance** | | 84.1 | 90.8 |
| **Fine-tuned** | NumNet+ V2 | 30.60 | 40.10 |
| | TagOp | 33.70 | 42.50 |
| | MHST | 41.50 | 50.70 |
| | Doc2SoarGraph | 59.20 | 67.60 |
| **Zero-shot** | Vicuna (7B) | 28.44 | 36.72 |
| | LLaMA 2-Chat (7B) | 34.52 | 42.32 |
| | MAmmoTH (70B) | 35.42 | 42.82 |
| | WizardMath (70B) | 36.44 | 41.55 |
| | LLaMA 2-Chat (70B) | 41.91 | 49.74 |
| | GPT3.5-Turbo | 52.74 | 61.40 |
| | GPT-4 | 64.46 | 72.20 |
| **Ours** | TAT-LLM (7B) | (+4.99) **69.45** | (+5.55) **77.75** |

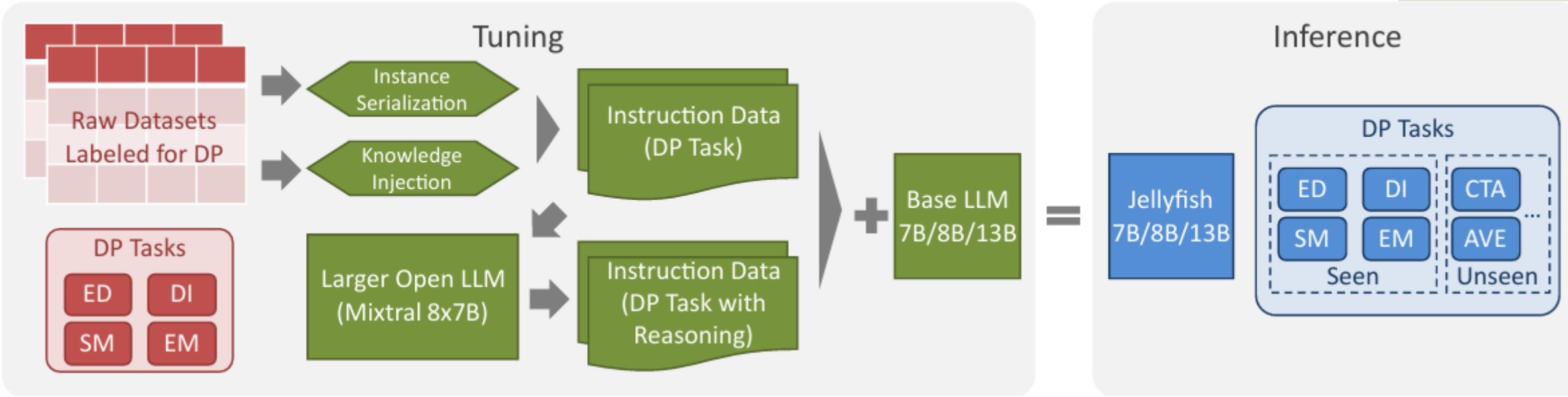
Table 4: Performance of our TAT-LLM model and compared models on the test set of TAT-DQA.

# Jellyfish

Table Preprocessing

OpenOrca-Platypus2-13B
Llama-3.1-8B-Instruct
Mistral-7B-Instruct-v0.2

- Specialized in Data Preprocessing
- Data Preprocessing Interpreter
- Jellyfish Reasoning Dataset



# Jellyfish

| Task | Type | Dataset | Model | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Best of non-LLM | GPT-3 | GPT-3.5 | GPT-4 | GPT-4o | Table-GPT | Jellyfish-7B | Jellyfish-8B | Jellyfish-13B |
| ED | Seen | Adult | 99.10 | 99.10 | 92.01 | 92.01 | 83.58 | – | 77.40 | 73.74 | **99.33** |
| | | Hospital | 94.40 | **97.80** | 90.74 | 90.74 | 44.76 | – | 94.51 | 93.40 | 95.59 |
| | Unseen | Flights | 81.00 | – | – | **83.48** | 66.01 | – | 69.15 | 66.21 | 82.52 |
| | | Rayyan | 79.00 | – | – | 81.95 | 68.53 | – | 75.07 | 81.06 | **90.65** |
| DI | Seen | Buy | 96.50 | 98.50 | 98.46 | **100** | **100** | – | 98.46 | 98.46 | **100** |
| | | Restaurant | 77.20 | 88.40 | 94.19 | **97.67** | 90.70 | – | 89.53 | 87.21 | 89.53 |
| | Unseen | Flipkart | 68.00 | – | – | **89.94** | 83.20 | – | 87.14 | 87.48 | 81.68 |
| | | Phone | 86.70 | – | – | **90.79** | 86.78 | – | 86.52 | 85.68 | 87.21 |
| SM | Seen | MIMIC-III | 20.00 | – | – | 40.00 | 29.41 | – | **53.33** | 45.45 | 40.00 |
| | | Synthea | 38.50 | 45.20 | 57.14 | **66.67** | 6.56 | – | 55.56 | 47.06 | 56.00 |
| | Unseen | CMS | 50.00 | – | – | 19.35 | 22.22 | – | 42.86 | 38.10 | **59.29** |
| EM | Seen | Amazon-Google | 75.58 | 63.50 | 66.50 | 74.21 | 70.91 | 70.10 | **81.69** | 81.42 | 81.34 |
| | | Beer | 94.37 | **100** | 96.30 | **100** | 90.32 | 96.30 | **100.00** | **100.00** | 96.77 |
| | | DBLP-ACM | **98.99** | 96.60 | 96.99 | 97.44 | 95.87 | 93.80 | 98.65 | 98.77 | 98.98 |
| | | DBLP-GoogleScholar | 95.70 | 83.80 | 76.12 | 91.87 | 90.45 | 92.40 | 94.88 | 95.03 | **98.51** |
| | | Fodors-Zagats | **100** | **100** | **100** | **100** | 93.62 | **100** | **100** | **100** | **100** |
| | | iTunes-Amazon | 97.06 | 98.20 | 96.40 | **100** | 98.18 | 94.30 | 96.30 | 96.30 | 98.11 |
| | Unseen | Abt-Buy | 89.33 | – | – | **92.77** | 78.73 | – | 86.06 | 88.84 | 89.58 |
| | | Walmart-Amazon | 86.89 | 87.00 | 86.17 | **90.27** | 79.19 | 82.40 | 84.91 | 85.24 | 89.42 |
| Average | | | 80.44 | - | - | 84.17 | 72.58 | - | 82.74 | 81.55 | **86.02** |

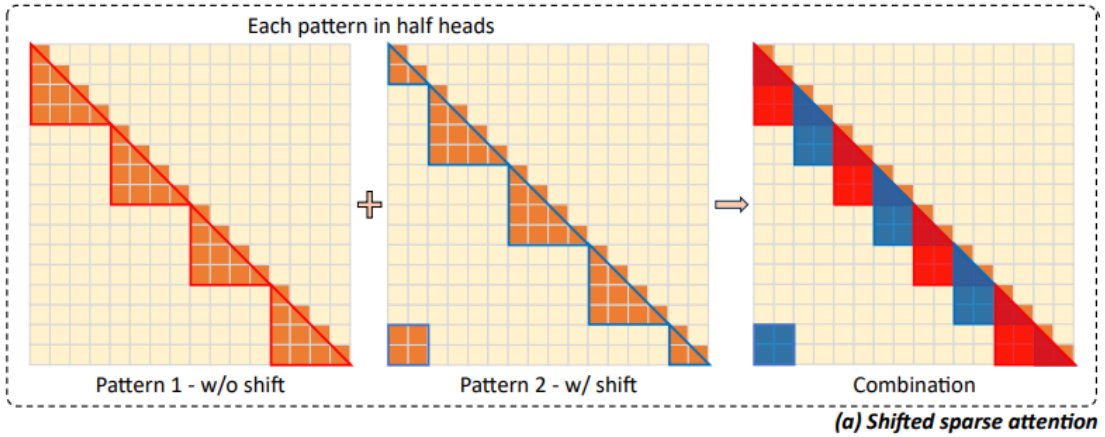
| Task | Dataset | Model | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | RoBERTa (159 shots) | RoBERTa (356 shots) | Stable Beluga 2 70B | SOLAR 70B | GPT-3.5 | GPT-4 | GPT-4o | Jellyfish-7B | Jellyfish-8B | Jellyfish-13B |
| CTA | SOTAB | 79.20 | 89.73 | – | – | 89.47 | **91.55** | 65.06 | 83.00 | 76.33 | 82.00 |
| AVE | AE-110k | – | – | 52.10 | 49.20 | **61.30** | 55.50 | 55.77 | 56.09 | 59.55 | 58.12 |
| | OA-Mine | – | – | 50.80 | 55.20 | 62.70 | **68.90** | 60.20 | 51.98 | 59.22 | 55.96 |

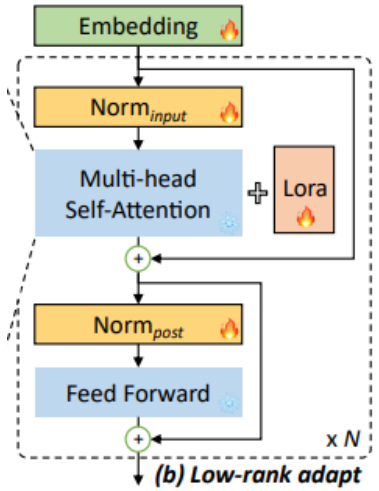
# LongLoRA

- Extends the context sizes of large language models with limited computational cost
- Shifted Sparse Attention (S2-Attn)

- Advantages
  - Efficient Context Extension
  - Compatible with existing optimizations

- Disadvantages
  - Requires careful handling of attention mechanisms



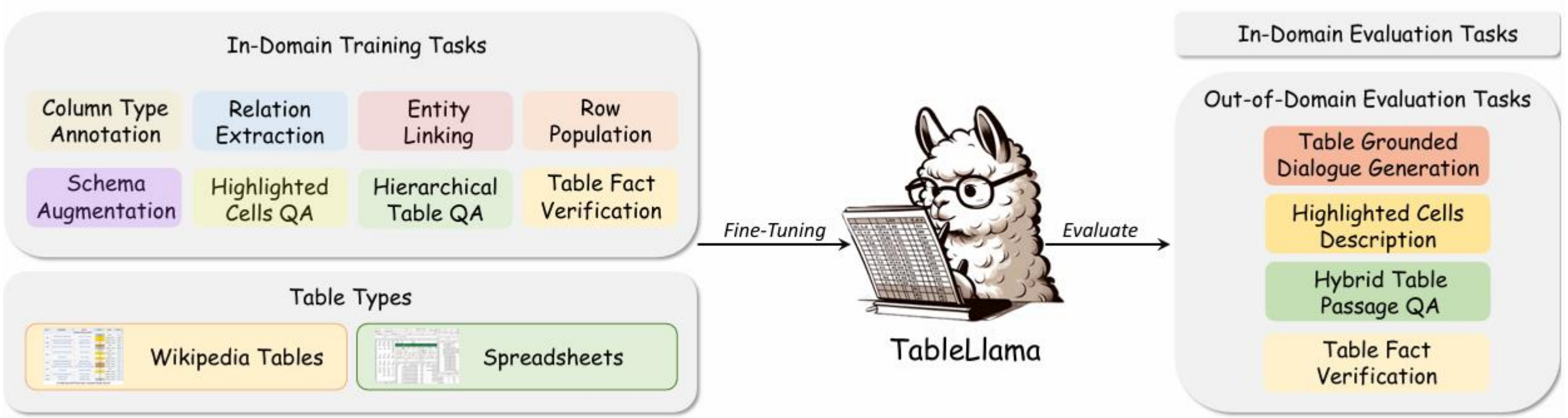

Chen et al. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models. ICLR 2024

# TableLlama

Table Preprocessing | Table Understanding | Table Analysis

Llama2-7B

- TableInstruct Dataset
- Real-world data sources
- LongLoRA for Extending Context Window




Li et al. TableLlama: Towards Open Large Generalist Models for Tables. NAACL 2024

# TableLlama

| Task Category | Task Name | Dataset | In-domain | #Train (Table/Sample) | #Test (Table/Sample) | Input Token Length min | max | median |
|---|---|---|---|---|---|---|---|---|
| Table Interpretation | Col Type Annot. | TURL (Deng et al., 2020) | Yes | 397K/628K | 1K/2K | 106 | 8192 | 2613 |
| | Relation Extract. | | Yes | 53K/63K | 1K/2K | 2602 | 8192 | 3219 |
| | Entity Linking | | Yes | 193K/1264K | 1K/2K | 299 | 8192 | 4667 |
| Table Augmentation | Schema Aug. | TURL (Deng et al., 2020) | Yes | 288K/288K | 4K/4K | 160 | 1188 | 215 |
| | Row Pop. | | Yes | 286K/286K | 0.3K/0.3K | 264 | 8192 | 1508 |
| Question Answering | Hierarchical Table QA | HiTab (Cheng et al., 2022b) | Yes | 3K/7K | 1K/1K | 206 | 5616 | 978 |
| | Highlighted Cells QA | FeTaQA (Nan et al., 2022) | Yes | 7K/7K | 2K/2K | 261 | 5923 | 740 |
| | Hybrid Table QA | HybridQA (Chen et al., 2020b) | No | – | 3K/3K | 248 | 2497 | 675 |
| | Table QA | WikiSQL (Zhong et al., 2017) | No | – | 5K/16K | 198 | 2091 | 575 |
| | Table QA | WikiTQ (Pasupat and Liang, 2015) | No | – | 0.4K/4K | 263 | 2688 | 709 |
| Fact Verification | Fact Verification | TabFact (Chen et al., 2020a) | Yes | 16K/92K | 2K/12K | 253 | 4975 | 630 |
| | | FEVEROUS (Aly et al., 2021) | No | – | 4K/7K | 247 | 8192 | 648 |
| Dialogue Generation | Table Grounded Dialogue Generation | KVRET (Eric et al., 2017) | No | – | 0.3K/0.8K | 187 | 1103 | 527 |
| Data-to-Text | Highlighted Cells Description | ToTTo (Parikh et al., 2020) | No | – | 7K/8K | 152 | 8192 | 246 |

Li et al. TableLlama: Towards Open Large Generalist Models for Tables. NAACL 2024

# TableLlama

| In-domain Evaluation | | | | | | |
|---|---|---|---|---|---|---|
| Datasets | Metric | Base | TableLlama | SOTA | GPT-3.5 | GPT-4§ |
| Column Type Annotation | F1 | 3.01 | 94.39 | **94.54***† (Deng et al., 2020) | 30.88 | 31.75 |
| Relation Extraction | F1 | 0.96 | 91.95 | **94.91***† (Deng et al., 2020) | 27.42 | 52.95 |
| Entity Linking | Accuracy | 31.80 | **93.65** | 84.90*† (Deng et al., 2020) | 72.15 | 90.80 |
| Schema Augmentation | MAP | 36.75 | **80.50** | 77.55*† (Deng et al., 2020) | 49.11 | 58.19 |
| Row Population | MAP | 4.53 | 58.44 | **73.31***† (Deng et al., 2020) | 22.36 | 53.40 |
| HiTab | Exec Acc | 14.96 | **64.71** | 47.00*† (Cheng et al., 2022a) | 43.62 | 48.40 |
| FeTaQA | BLEU | 8.54 | **39.05** | 33.44 (Xie et al., 2022) | 26.49 | 21.70 |
| TabFact | Accuracy | 41.65 | 82.55 | **84.87*** (Zhao and Yang, 2022) | 67.41 | 74.40 |

| Out-of-domain Evaluation | | | | | | | |
|---|---|---|---|---|---|---|---|
| Datasets | Metric | Base | TableLlama | SOTA | $\Delta_{Base}$ | GPT-3.5 | GPT-4§ |
| FEVEROUS | Accuracy | 29.68 | 73.77 | 85.60 (Tay et al., 2022) | +44.09 | 60.79 | 71.60 |
| HybridQA | Accuracy | 23.46 | 39.38 | 65.40* (Lee et al., 2023) | +15.92 | 40.22 | 58.60 |
| KVRET | Micro F1 | 38.90 | 48.73 | 67.80 (Xie et al., 2022) | +9.83 | 54.56 | 56.46 |
| ToTTo | BLEU | 10.39 | 20.77 | 48.95 (Xie et al., 2022) | +10.38 | 16.81 | 12.21 |
| WikiSQL | Accuracy | 15.56 | 50.48 | 92.70 (Xu et al., 2023b) | +34.92 | 41.91 | 47.60 |
| WikiTQ | Accuracy | 29.26 | 35.01 | 57.50† (Liu et al., 2022) | +5.75 | 53.13 | 68.40 |

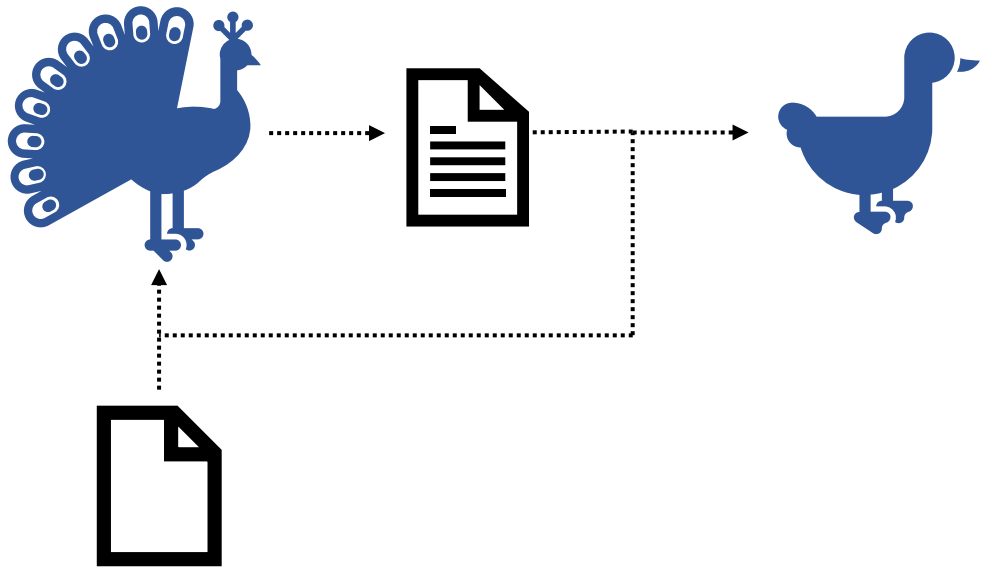
# Knowledge Distillation

- Training a **student model** using the output of a **teacher model**

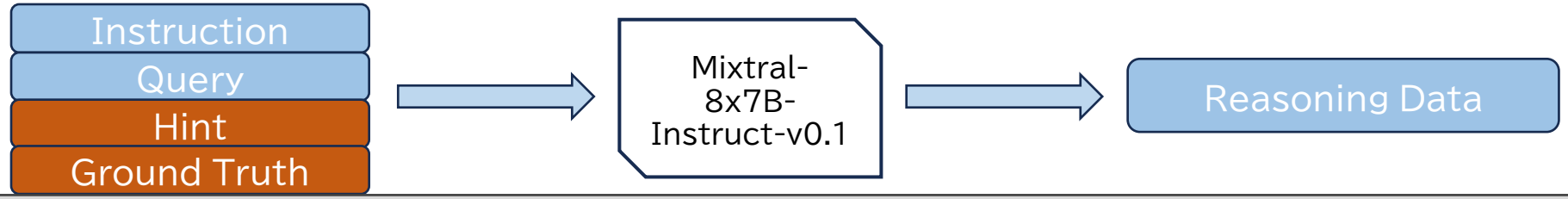
- Advantages
  - Model Compression
  - Enhanced Generalization

- Disadvantages
  - Rely on the teacher model
  - Potential Performance Trade-off

- Use Case
  - Lake of labeled training data
  - Smaller model is required



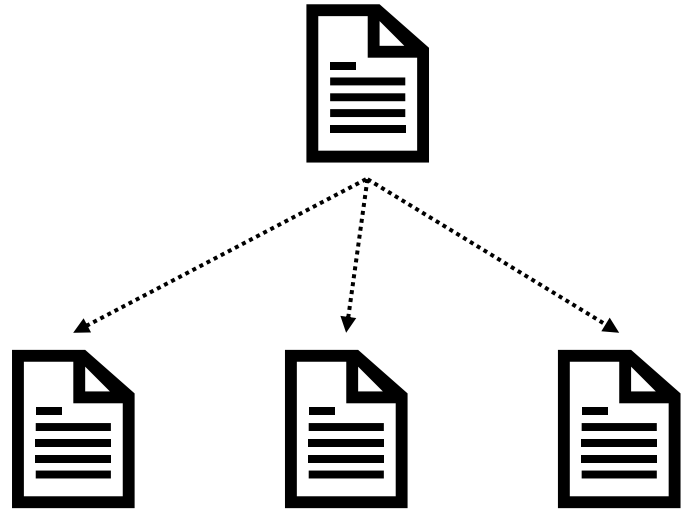
# Jellyfish




**(system message)** You are an AI assistant that follows instruction extremely well. User will give you a question. Your task is to answer as faithfully as you can. While answering, provide detailed explanation and justify your answer.
**(task description)** You are tasked with determining whether two Products listed below are the same based on the information provided. Carefully compare all the attributes before making your decision.
**(injected knowledge)** Note that missing values (N/A or "nan") should not be used as a basis for your decision. Note that different factories can belong to the same parent company. The company name of Product B may occur in its product name.
**(instance content)** Product A: [name: "Sequoia American Amber Ale", factory: "Wig And Pen"] Product B: [name: "Aarhus Cains Triple A American Amber Ale", factory: "Aarhus Bryghus"]
**(question)** Are Product A and Product B the same Product?
**(output format)** After your reasoning, finish your response in a separate line with and ONLY with your final answer. Choose your final answer from [Yes, No].
**(answer hint)** You can use the "Hint" below, but your response cannot contain any information from it. Hint: the final answer is "No"

# Data Augmentation Enhanced Fine-Tuning

- Incorporates data augmentation to enrich the training dataset
- Advantages
  - Improved Generalization
  - Enhanced Robustness
  - Data Efficiency
- Disadvantages
  - Potential for Mislabeling
  - Diminishing Returns
- Use Case
  - Limited Data Scenarios
  - Overfitting Issues



# Table-GPT

- Synthesize Diverse Table-Tasks
  - Task Synthesis for Task-Diversity
    - Creating new, diverse tasks from real-world table data
  - Data-Diversity through Test Case Synthesis
    - Generate multiple test cases for a broader range of table scenarios

**Algorithm 1:** Synthesize table-tasks for table-tuning

**input** : A corpus of diverse real tables **C**, a set of table-task types **S**
**output** : Diverse synthesized table-tasks $A = \{(Ins, T, C)\}$

$D \leftarrow \{\}, A \leftarrow \{\}$
**foreach** $T \in \mathbf{C}, S \in \mathbf{S}$ **do**
  $(Ins, T, C) \leftarrow$ Synthesize-Table-Task$(S, T)$ // (Section 4.2)
  $D \leftarrow D \cup (Ins, T, C)$
**foreach** $(Ins, T, C) \in D$ **do**
  $Ins' \leftarrow$ Augment-Instruction$(Ins)$ // (Section 4.3)
  $T' \leftarrow$ Augment-Table$(T)$ // (Section 4.3)
  $C' \leftarrow$ Augment-Completion$(C)$ // (Section 4.3)
  $A \leftarrow A \cup (Ins', T', C')$
**return** $A$

# Table-GPT

- Augment Synthesized Table-Tasks
  - Instruction-Level Augmentation
  - Table-Level Augmentation
  - Completion-Level Augmentation

**T-8: Entity Matching**

**Instruction**: Please determine whether the first and second row below may refer to the same entity. Please answer "yes/no" in JSON.

**Row-1**:
| Brand | Model | Price | Description | ...
| Apple | iPhone-15 | $995 | ... | ...
**Row-2**:
| Brand | Model | Price | Description | ...
| Apple | iPhone pro | $1500 | ... | ...

**Model response**:
The first and second product are likely different, because they have very different prices. Therefore, the answer is **{"answer": "No"}.**

**T-9: Error Detection (misspellings)**

**Instruction**: Please examine the table below and check which cell may be misspelled. If there is no erroneous cell, return 'None'. Please answer using JSON: {"answer": ""}.

**Table**:
| State | Capital | Population | ...
| Washington | Olympia | 7.7 million | ...
| Missisipi | Jackson | 2.9 million | ...

**Model response**:
The cell " Missisipi" is misspelled, and should be corrected as " Mississippi". Therefore, the answer is **{"answer": "Missisipi"}.**

# Table Data Representation & Prompt Design

# Table-GPT

**Header Value Matching Zero-Shot**

**Prompt:**

```
Task: Given the input table data and the list of candidate headers, please determine the most suitable column
    header for each column in the table. Please only choose column headers from the candidate list. Please only
    return the most suitable column header for each column. Return the chosen column headers in a list. Do not
    return the entire table. Return the final result as JSON in the format {"column_headers": "<a list of
    headers for each column chosen from the candidate list>"}.

[Q]:
**Table Data:**
||||
|---|---|---|
|1|681|Multihull|
|3|911|Keelboat|
|2|947|Dinghy|
|nan|920|Dinghy|
|1|870|Dinghy|

**Candidate column headers:**
- Portsmouth Number
- crew
- Type

Return the final result as JSON in the format {"column_headers": "<a list of headers for each column chosen from
     the candidate list>"}.
[A]:
```

**Completion:**

```
{"column_headers": ["crew", "Portsmouth Number", "Type"]}
```

Li et al. Table-GPT: Table-tuned GPT for Diverse Table Tasks. SIGMOD 2024

# SpreadSheetLLM

Prompt Template for Spreadsheet Table Detection

```
(INSTRUCTION:) Given an input that is a string denoting data of cells in an Excel
spreadsheet. The input spreadsheet contains many tuples, describing the cells with
content in the spreadsheet. Each tuple consists of two elements separated by a '|':
the cell content and the cell address/region, like (Year|A1), ( |A1) or(IntNum|A1:B3).
The content in some cells such as '#,##0'/'d-mmm-yy'/'H:mm:ss',etc., represents the
CELL DATA FORMATS of Excel. The content in some cells such as
'IntNum'/'DateData'/'EmailData',etc., represents a category of data with the same for
mat and similar semantics. For example, 'IntNum' represents integer type data, and
'ScientificNum' represents scientific notation type data. 'A1:B3' represents a region
in a spreadsheet, from the first row to the third row and from column A to column B.
Some cells with empty content in the spreadsheet are not entered. Now you should tell
me the range of the table in a format like A2:D5, and the range of the table should
only CONTAIN HEADER REGIONandthedata region. DON'T include the title or comments. Note
that there can be more than one table in a string, so you should return all the RANGE.
DON'TADDOTHERWORDSOREXPLA
NATION.
(INPUT) [Encoded Spreadsheet]
```

Tian et al. SpreadSheetLLM: Encoding Spreadsheets for Large Language Models. arXiv 2024

# TAT-LLM

Template for FinQA dataset following Step-wise Pipeline.

```
Below is an instruction that describes a question answering task in the finance
domain, paired with an input table and its relevant text that provide further context.
The given question is relevant to the table and text. Generate an appropriate answer
to the given question.
(Instruction) Given a table and a list of texts in the following, what is the answer
to the question? Please complete the task in three steps:
(Extractor) In the first step, extract the relevant numerical values from the provided
table or texts. Store these in the variable '{evidence}'. If there are multiple
values, separate them using the '#' symbol.
(Reasoner) In the second step, generate an equation using the extracted numerical
values. Store this equation in the variable
 '{equation}'.
(Executor:) In the third step, calculate the answer based on the equation and store it
in the variable '{answer}'. Please organize the results in the following table:
 | step | output |
 | 1 | {evidence} |
 | 2 | {equation} |
 | 3 | {answer} |
 Finally, present the calculated answer in the format: "The answer is: {answer}"
(Table) (Text) (Question)
```

# Jellyfish

Direct and reasoning prompt Template for entity matching

| | DP Task Data | DP Task with Reasoning Data |
|---|---|---|
| system message | You are an AI assistant that follows instruction extremely well. User will give you a question. Your task is to answer as faithfully as you can. | You are an AI assistant that follows instruction extremely well. User will give you a question. Your task is to answer as faithfully as you can. While answering, provide detailed explanation and justify your answer. |
| task description | You are tasked with determining whether two Products listed below are the same based on the information provided. Carefully compare all the attributes before making your decision. | |
| injected knowledge | Note that missing values (N/A or "nan") should not be used as a basis for your decision. | |
| instance content | Product A: [name: "Sequoia American Amber Ale", factory: "Wig And Pen"]<br>Product B: [name: "Aarhus Cains Triple A American Amber Ale", factory: "Aarhus  Bryghus"] | |
| question | Are Product A and Product B the same Product? | |
| output format | Choose your answer from: [Yes, No] | After your reasoning, finish your response in a separate line with and ONLY with your final answer. Choose your final answer from [Yes, No]. |

# TableLlama

**### Instruction:**
This is a table **row population** task. The goal of this task is to populate the possible entities of the selected column for a table, given the Wikipedia page title, ... You will be given a list of entity candidates. Please rank them so that the most likely entities come first.

**### Input:**
[TLE] The Wikipedia page is about NBA conference finals. The Wikipedia section is about eastern conference finals. The table headers are: | year | champion | ... You need to populate the column: year. [SEED] The seed entity is <1971_NBA_playoffs>.

**### Question:**
The entity candidates are: <2003_NBA_playoffs>, <1982-83_Washington_Bullets_season>, <2004_NBA_playoffs>, <Philadelphia_76ers>, <1983-84_Washington_Bullets_season>, <1952_NBA_playoffs>, ...

**### Response:** <1972_NBA_playoffs>, <1973_NBA_playoffs>, <1974_NBA_playoffs>, <1975_NBA_playoffs>, <1976_NBA_playoffs>, ...

**### Instruction:**
This is a **hierarchical table question answering** task. The goal for this task is to answer the given question based on the given table. The table might be hierarchical.

**### Input:**
[TLE] The table caption is department of defense obligations for research, development, test, and evaluation, by agency: 2015-18. [TAB] | agency | 2015 | 2016 | ... [SEP] | department of defense | department of defense | ... [SEP] | rdt&e | 61513.5 | ... [SEP] | total research | 6691.5 | ... [SEP] | basic research | 2133.4 | ... [SEP] | defense advanced research projects agency | ...

**### Question:**
How many dollars are the difference for basic research of defense advanced research projects agency increase between 2016 and 2018?

**### Response:** 80.3.

Li et al. TableLlama: Towards Open Large Generalist Models for Tables. NAACL 2024

# Q&A

\Orchestrating a brighter world

NEC

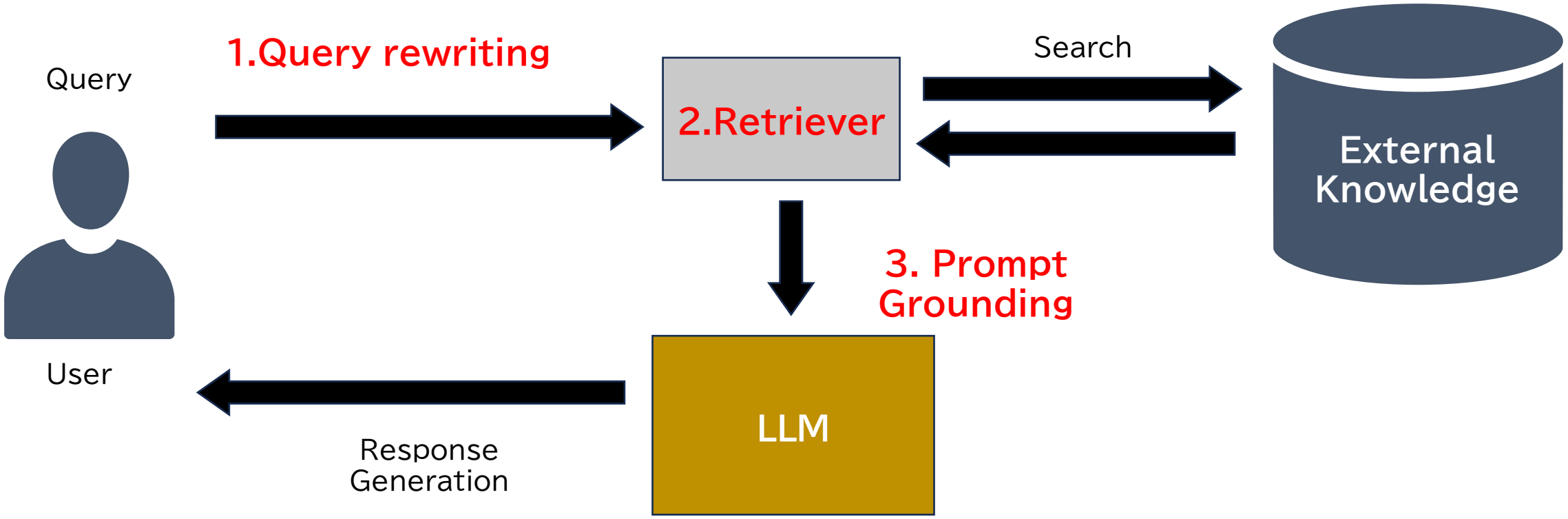
大阪大学
OSAKA UNIVERSITY

# On the Use of Large Language Models for Table Tasks

- Retrieval-augmented generation (RAG)

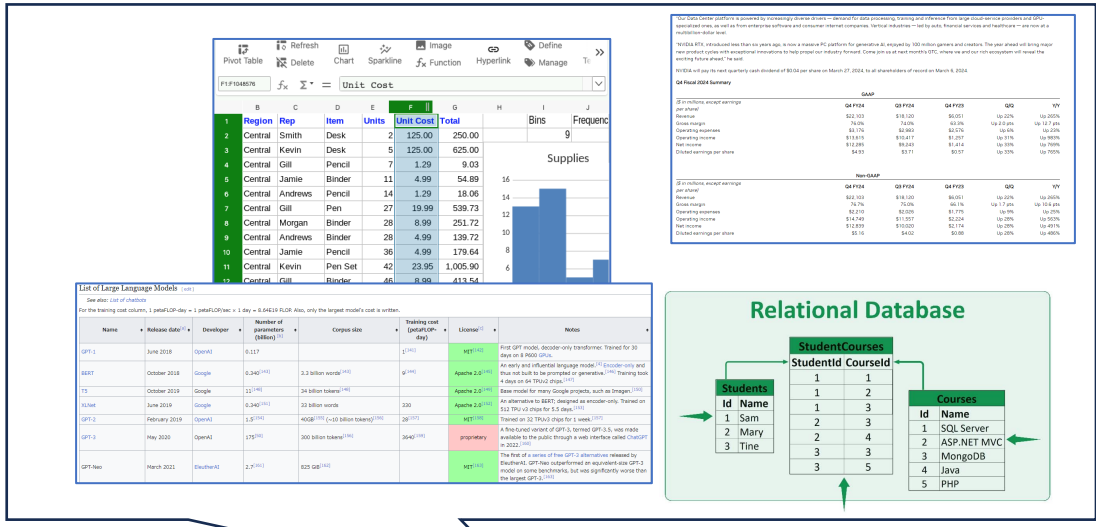
# What is RAG?

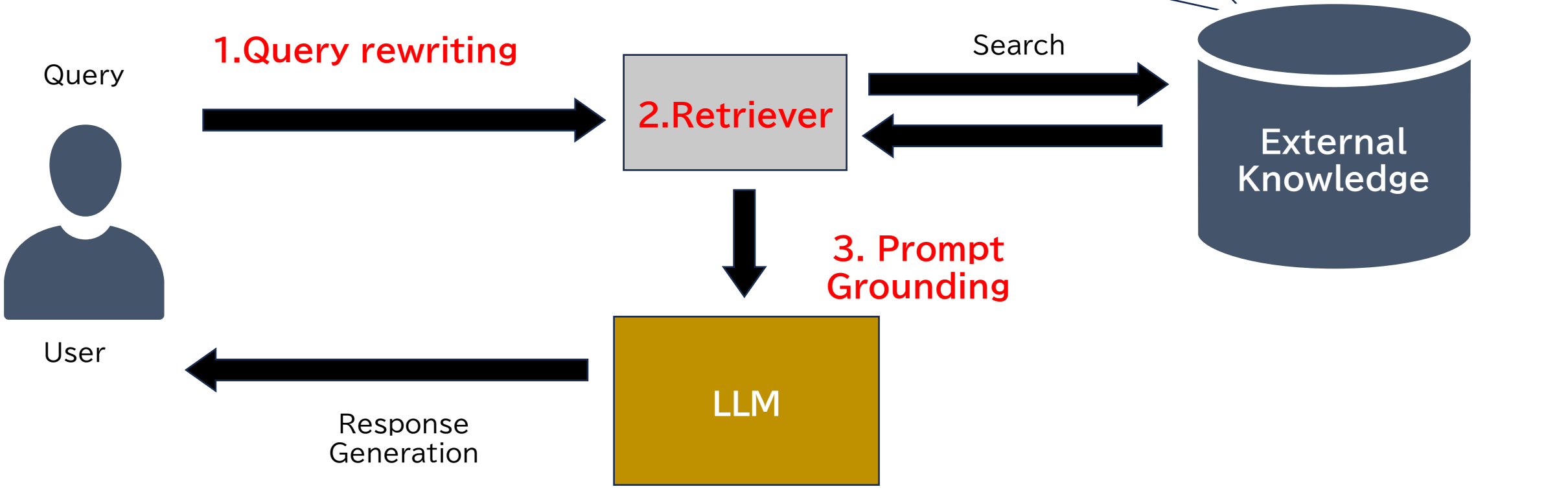
- LLM can not be trained frequently
- RAG is an efficient way to on-demand get external knowledge
  - Up-to-date knowledge
  - Less hallucination with retrieved context



# Motivation of RAG with table

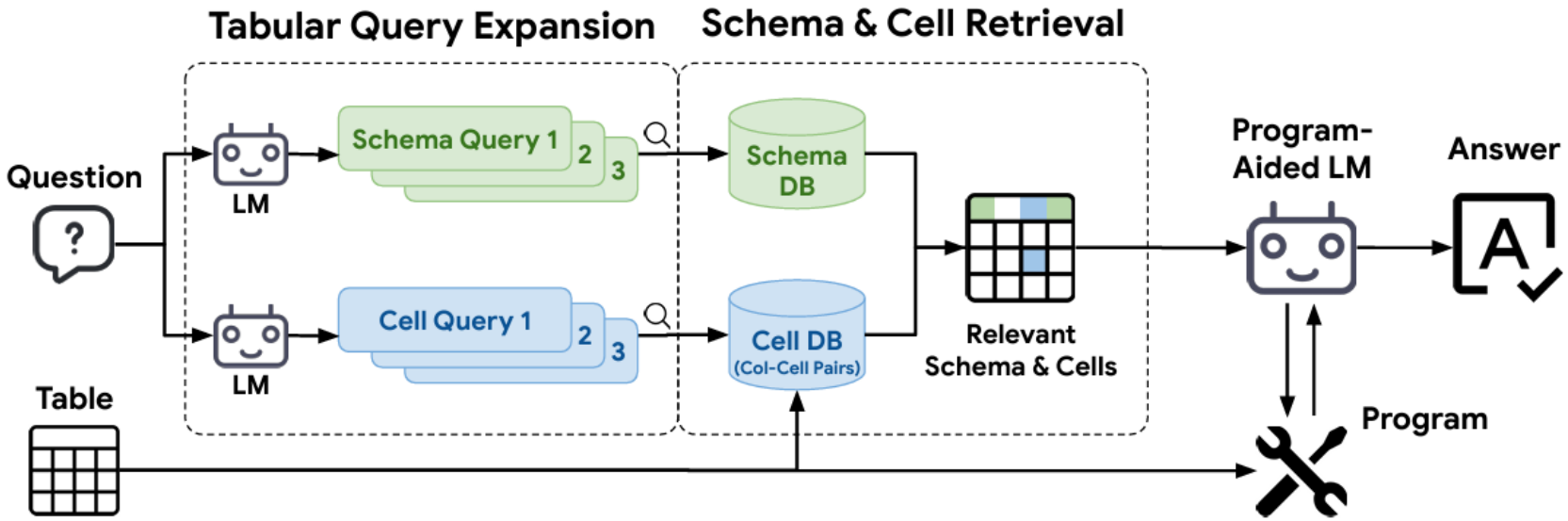
- **Knowledge is also in tables!**



# Query rewriting & Prompt Grounding

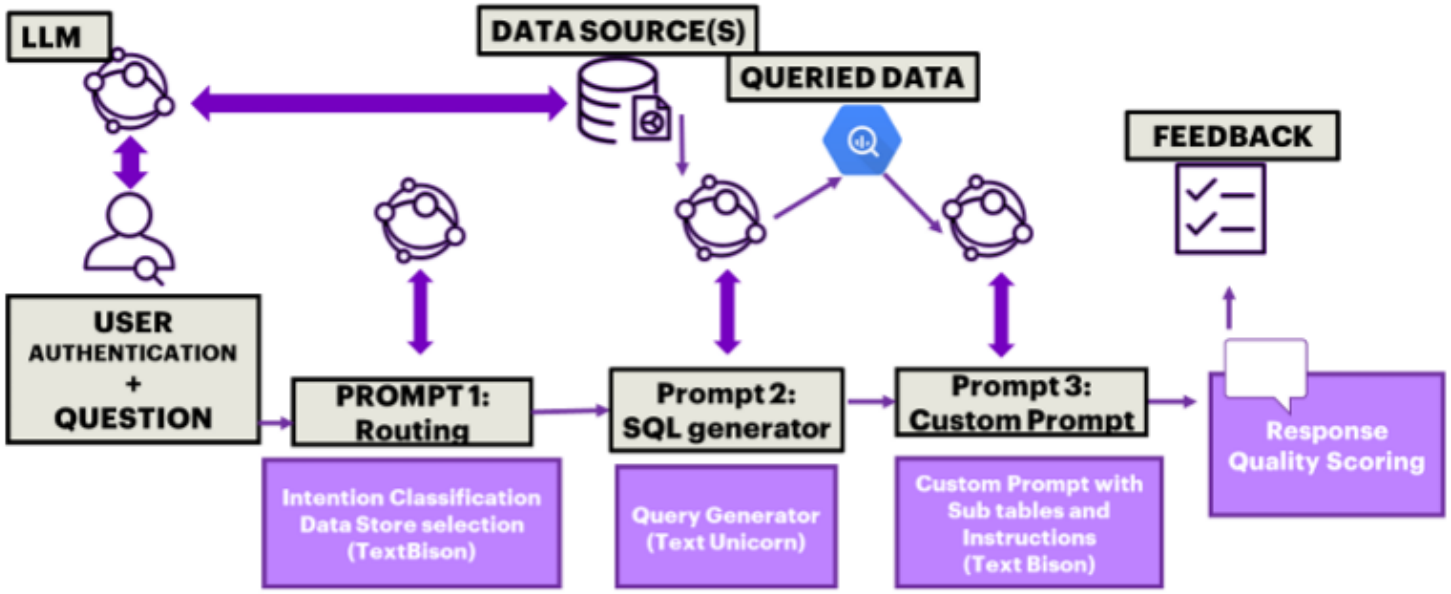
- TableRAG [arxiv24]: for table QA
  - Query rewriting
    - NL query to Schema Query and Cell Query
    - NL query to SQL query
  - Prompt Grounding
    - Python generation with <query, retrieved data>
      - Execute with python shell
      - Observe result then run another round of generation until getting final answer




https://arxiv.org/pdf/2410.04739

# Query rewriting & Prompt Grounding

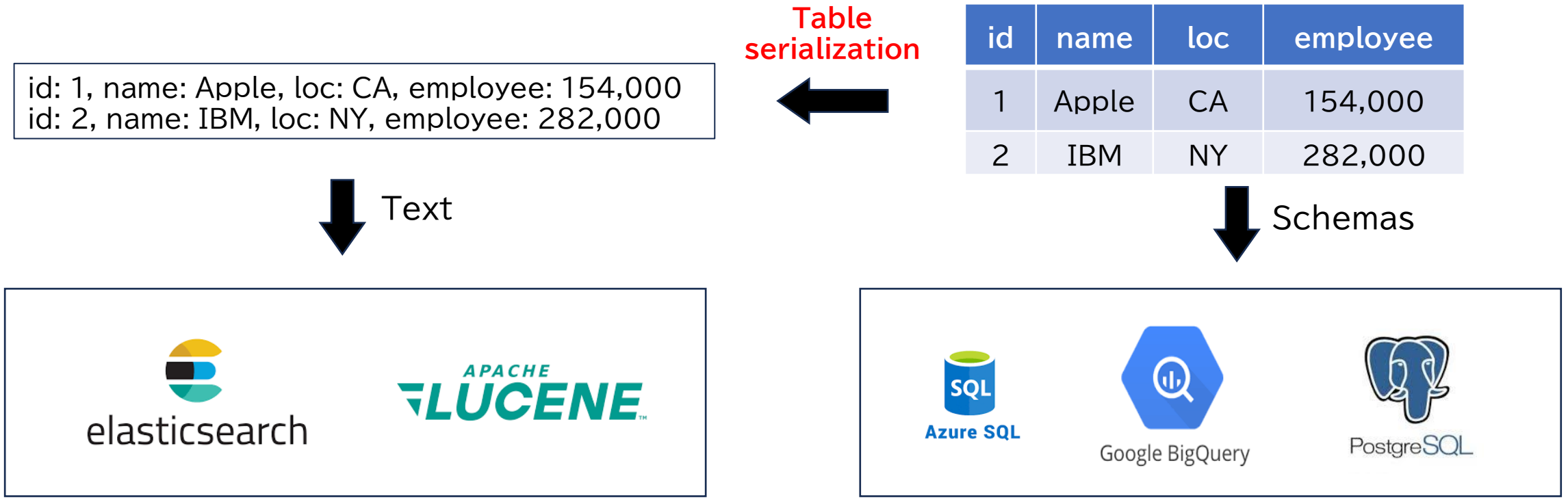
- ERATTA [BigData24]: for table QA
  - Query rewriting
    - Routing: select target table
    - SQL generation: query to target table
  - Grounding
    - SQL generation with <query, retrieved data>
      - Execute with BigQuery
      - Result feedback with 5 checks (entity, number, query, Regurgitation, Modifier)




https://arxiv.org/abs/2405.03963

# Table Retriever

- **Traditional retriever**
  - Keyword-based (BM25) for text-based retriever
  - SQL for relational database.
    - ERATTA, TableRAG, DB-GPT[arxiv23] https://arxiv.org/abs/2312.17449



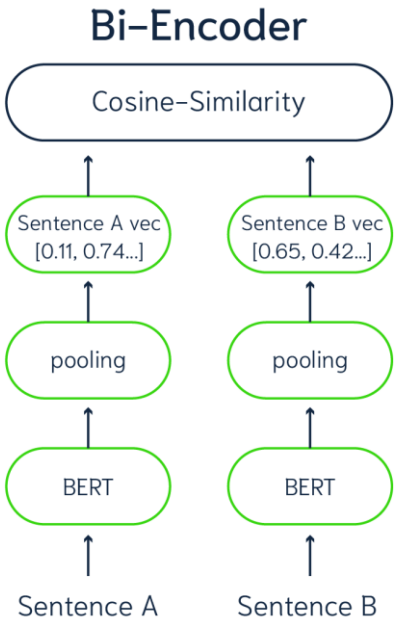
# Table Retriever

- **Dense Retriever: LM encoder + vector database**
  - Textual model: serialize table to text -> train text LM encoder

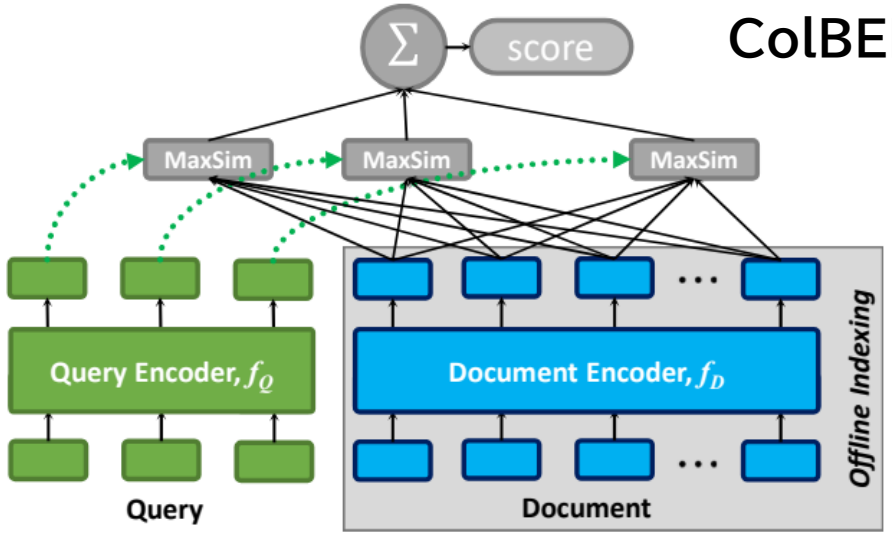
| | Model-architecture | serialization |
|---|---|---|
| T-RAG [arxiv22] | DPR | rows |
| Deep-join[VLDB23] | Sentence-BERT | columns |
| LI-RAGE [ACL23] | ColBERT | table |
| ITR [ACL23] | DPR | located row & col |

**DPR**

**SBERT**

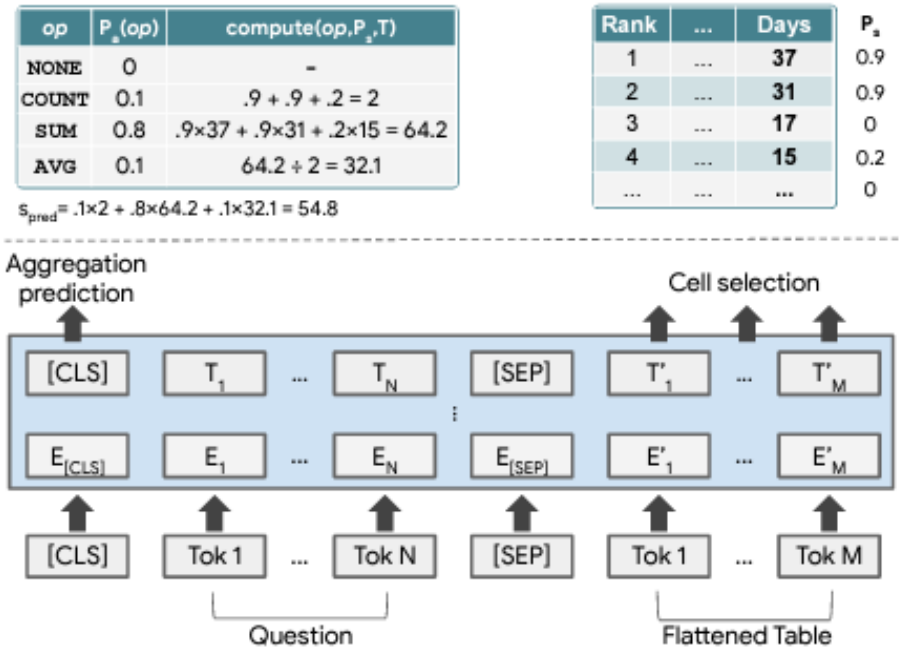



**ColBERT**

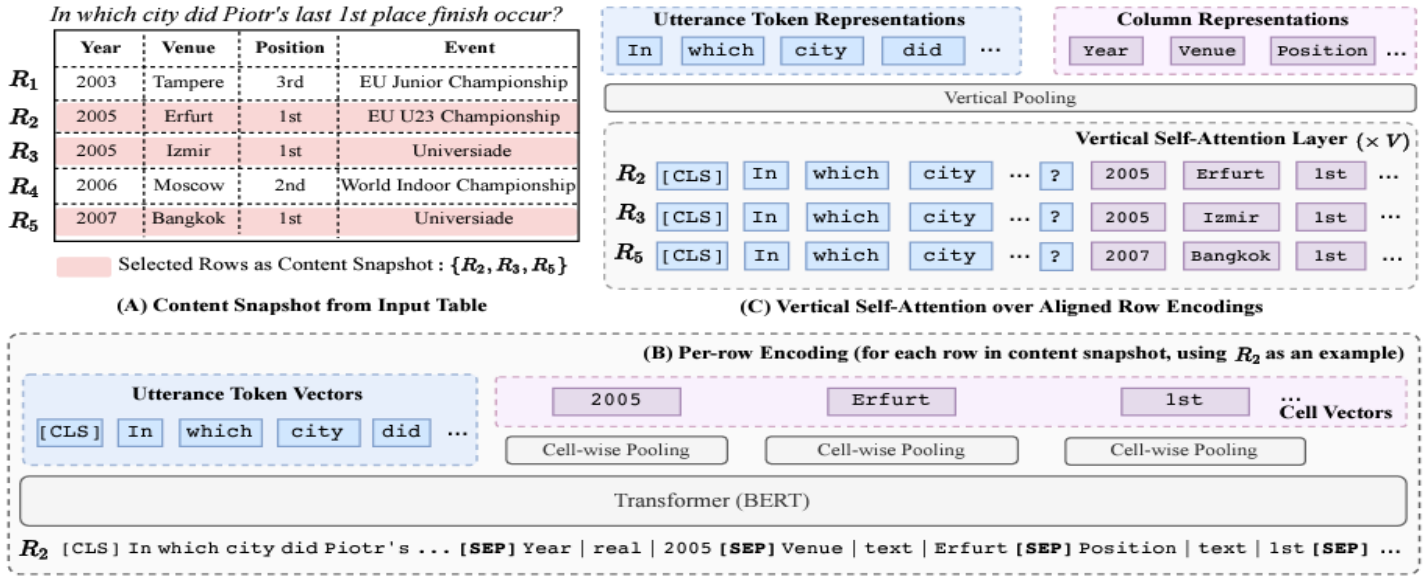


# Table Retriever

- **Dense Retriever: LM encoder + vector database**
  - Table-specific model: directly train the table with table-specific LM encoder
    - query -> text encoder
    - table -> table-specific encoder




TAPAS




TaBERT

# Table Retriever

- Do we need a table-specific model for table retriever?
- Table Retrieval May Not Necessitate Table-specific Model Design [ACL workshop22]
  - Table-specific(DTR) do not have significant improvement than text retrievers (DPR)
- Open-WikiTable [ACL23]
  - Same observation

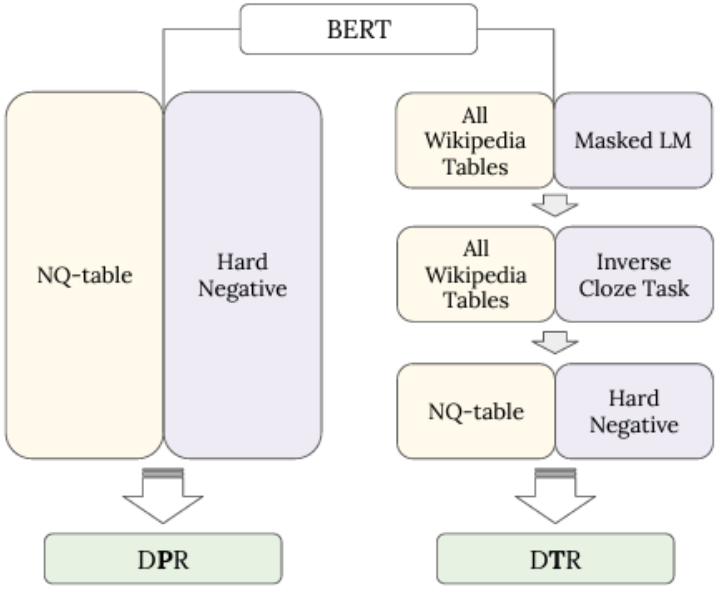



Figure 3: Comparison of DPR and DTR training.

| Model | Retrieval Accuracy @1 | @5 | @10 | @20 | @50 |
|---|---|---|---|---|---|
| DTR (medium) | 62.32 | 82.51 | 86.75 | 91.51 | 94.26 |
| DTR (large) | 63.98 | 84.27 | **89.65** | **93.48** | **95.65** |
| BERT-table | 60.97 | 79.81 | 85.51 | 88.20 | 91.62 |
| DPR | 57.04 | 80.54 | 86.13 | 89.54 | 92.34 |
| DPR-table | **67.91** | **84.89** | 88.72 | 90.58 | 92.86 |

NQ-tables

| Encoder Text | Encoder Table | Data | k=5 | k=10 | k=20 |
|---|---|---|---|---|---|
| BM25 | | Original | 6.6 | 8.0 | 10.3 |
| | | Decontextualized | 45.5 | 52.9 | 59.7 |
| | | Paraphrased | 42.2 | 48.9 | 56.1 |
| BERT | BERT | Original | 25.0 | 34.1 | 45.1 |
| | | Decontextualized | 91.6 | 96.0 | 97.8 |
| | | Paraphrased | 89.5 | 95.0 | 97.3 |
| BERT | TAPAS | Original | 19.4 | 28.1 | 38.5 |
| | | Decontextualized | 88.2 | 94.5 | 97.3 |
| | | Paraphrased | 84.0 | 91.4 | 95.6 |

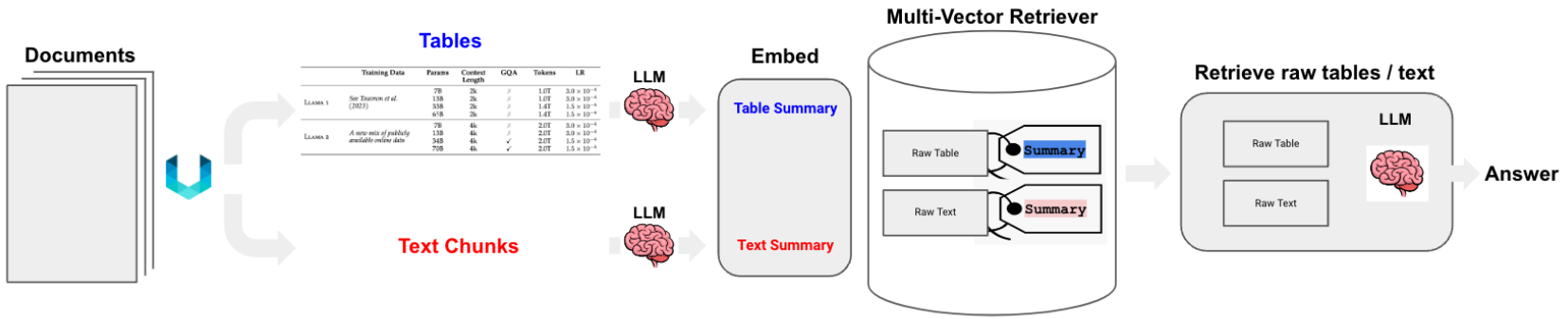
Open-WikitaTables

# Table Retriever (industry-view)

- **Entry: Keyword + traditional retriever**
  - Elasiticsearch: https://www.elastic.co/search-labs/blog/rag-with-llamaIndex-and-elasticsearch
  - RAG with Azure SQL: https://devblogs.microsoft.com/azure-sql/rag-with-azure-sql/
  - LlamaIndex: https://docs.llamaindex.ai/en/v0.10.18/api reference/indices/table.html

- **Advanced: LLM parse + textual embedding + vector database**

LangChain LlamaIndex




https://blog.langchain.dev/semi-structured-multi-modal-rag/
https://blog.langchain.dev/benchmarking-rag-on-tables/
https://docs.llamaindex.ai/en/stable/llama_cloud/llama_parse/



- **Deluxe (not recommended) : Human parse + textual embedding + vector database**

# Q&A

\Orchestrating a brighter world

NEC

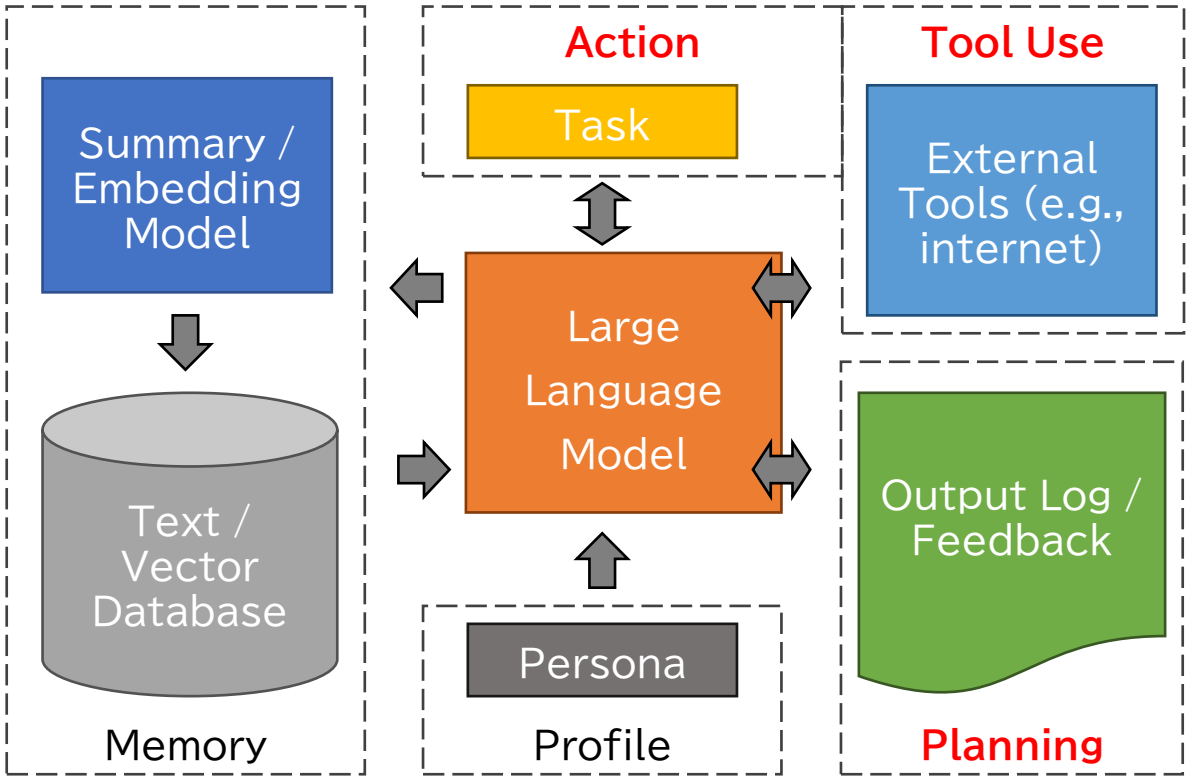
大阪大学
OSAKA UNIVERSITY

# On the Use of Large Language Models for Table Tasks

- LLM agents

# What is LLM agent

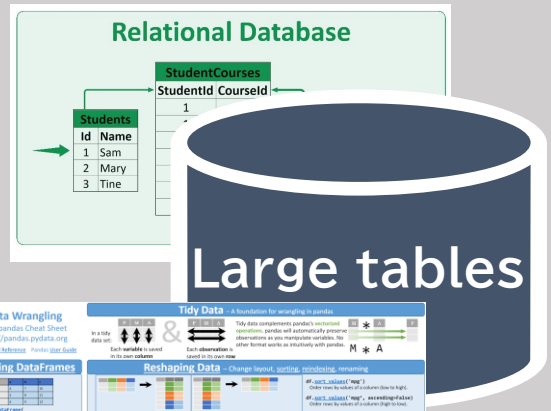
- Planning
  - Task decomposition
  - Task selection
  - Reflection & refinement
- Action & Tool use
  - Design an action space
  - Interactive with external tools
- Profile
  - Role play, prompt.
  - “You are an expert of table/SQL/Pandas…”
- Memory
  - Short-term memory (log in a dialog)
  - Long-term memory (RAG with outside knowledge)

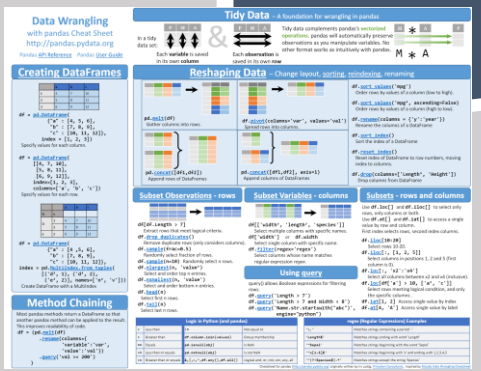



LLM Agent Components

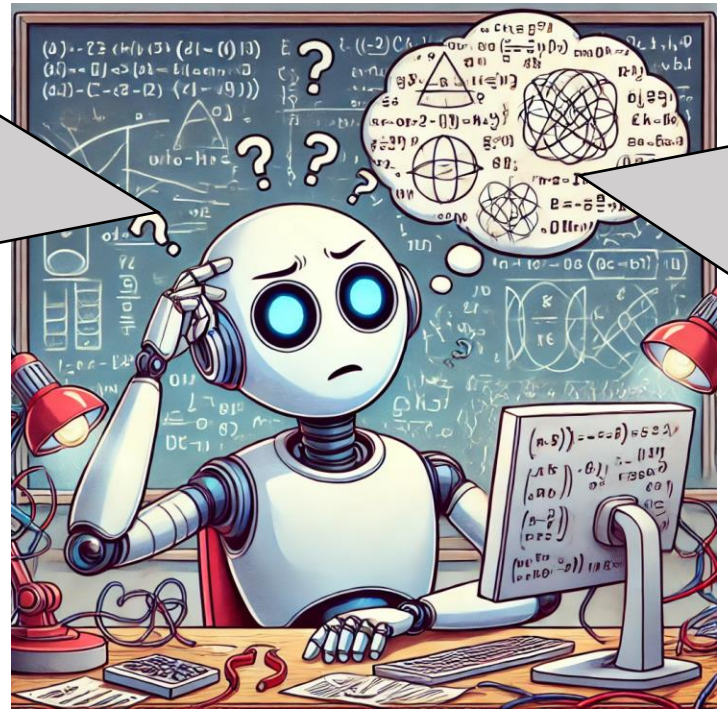
# Motivation of agent for table

- **One-time inference can not solve hard problems, need the “divide and concur”**
  - Table is complicate, only a part of contents is useful for a specific question.
  - Answer a question need multi-hop reasoning.
  - Need external tools
    - Numerical OLAP, aggregation, SQL, python, Pandas …




Understand & operate on large tables



| | |
|---|---|
| square beads | $2.97 per kilogram |
| oval beads | $3.41 per kilogram |
| flower-shaped beads | $2.18 per kilogram |
| star-shaped beads | $1.95 per kilogram |
| heart-shaped beads | $1.52 per kilogram |
| spherical beads | $3.42 per kilogram |
| rectangular beads | $1.97 per kilogram |

**Question:** If Tracy buys 5 kilograms of spherical beads, 4 kilograms of star-shaped beads, and 3 kilograms of flower-shaped beads, how much will she spend? (unit: $)
**Answer:** 31.44
**Solution:**
Find the cost of the spherical beads. Multiply: $3.42 × 5 = $17.10.
Find the cost of the star-shaped beads. Multiply: $1.95 × 4 = $7.80.
Find the cost of the flower-shaped beads. Multiply: $2.18 × 3 = $6.54.
Now find the total cost by adding: $17.10 + $7.80 + $6.54 = $31.44.
She will spend $31.44.

**Sandwich sales**

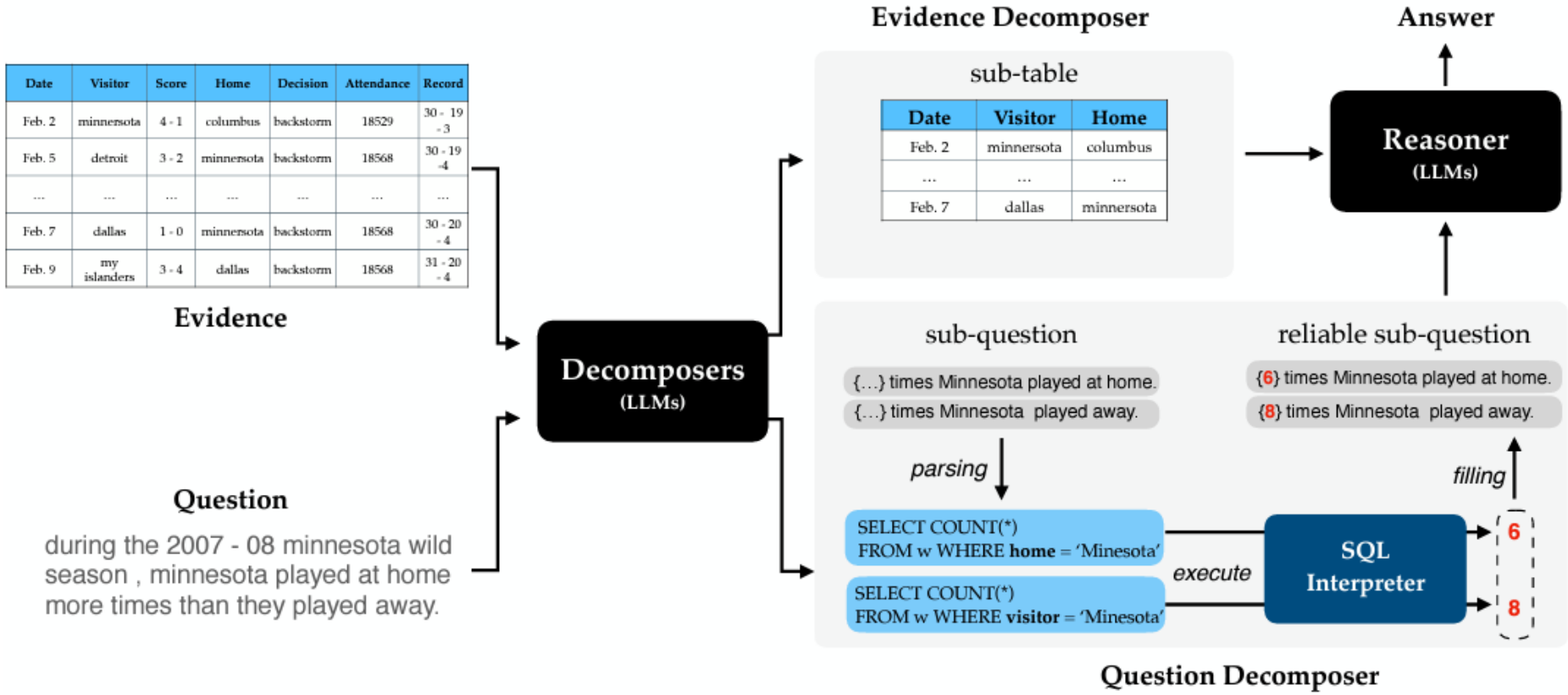
| Shop | Tuna | Egg salad |
|---|---|---|
| City Cafe | 6 | 5 |
| Sandwich City | 3 | 12 |
| Express Sandwiches | 7 | 17 |
| Sam's Sandwich Shop | 1 | 6 |
| Kelly's Subs | 3 | 4 |

**Question:** As part of a project for health class, Cara surveyed local delis about the kinds of sandwiches sold. Which shop sold fewer sandwiches, Sandwich City or Express Sandwiches?
**Options:** (A) Sandwich City (B) Express Sandwiches
**Answer:** (A) Sandwich City
**Solution:**
Add the numbers in the Sandwich City row. Then, add the numbers in the Express Sandwiches row.
Sandwich City: 3 + 12 = 15. Express Sandwiches: 7 + 17 = 24.
15 is less than 24. Sandwich City sold fewer sandwiches.

Mathematical reasoning from tables

# Planning:

- **Task decomposition**
  - Datar [SIGIR23]: Table QA
  - Question decomposition and table decomposition
    - sub question -> SQL -> query sub-table -> answer




https://arxiv.org/pdf/2301.13808

# Planning:

- **Task decomposition & Reflection**
  - DIN-SQL [NeurIPS23]: Text-to-SQL
  - Predefine the decomposition into 1 + 3 = 4 sub tasks
    - Schema linking -> Generate 3-levels SQL with 3 sub tasks
  - Self-correction the final answer

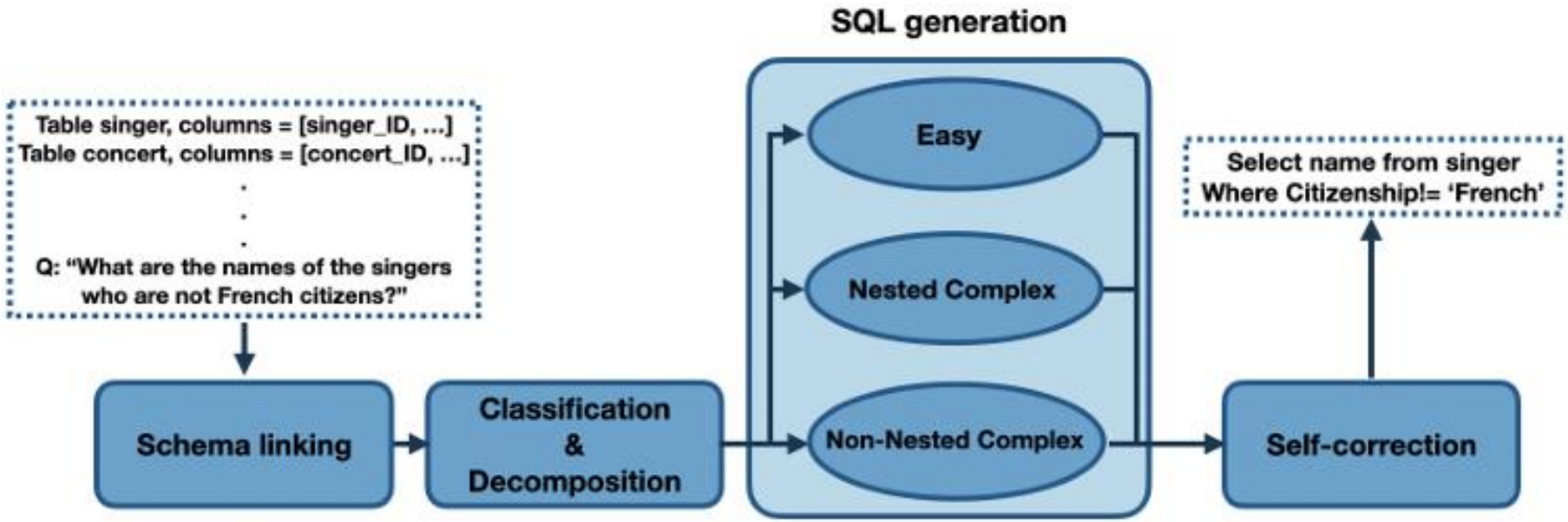
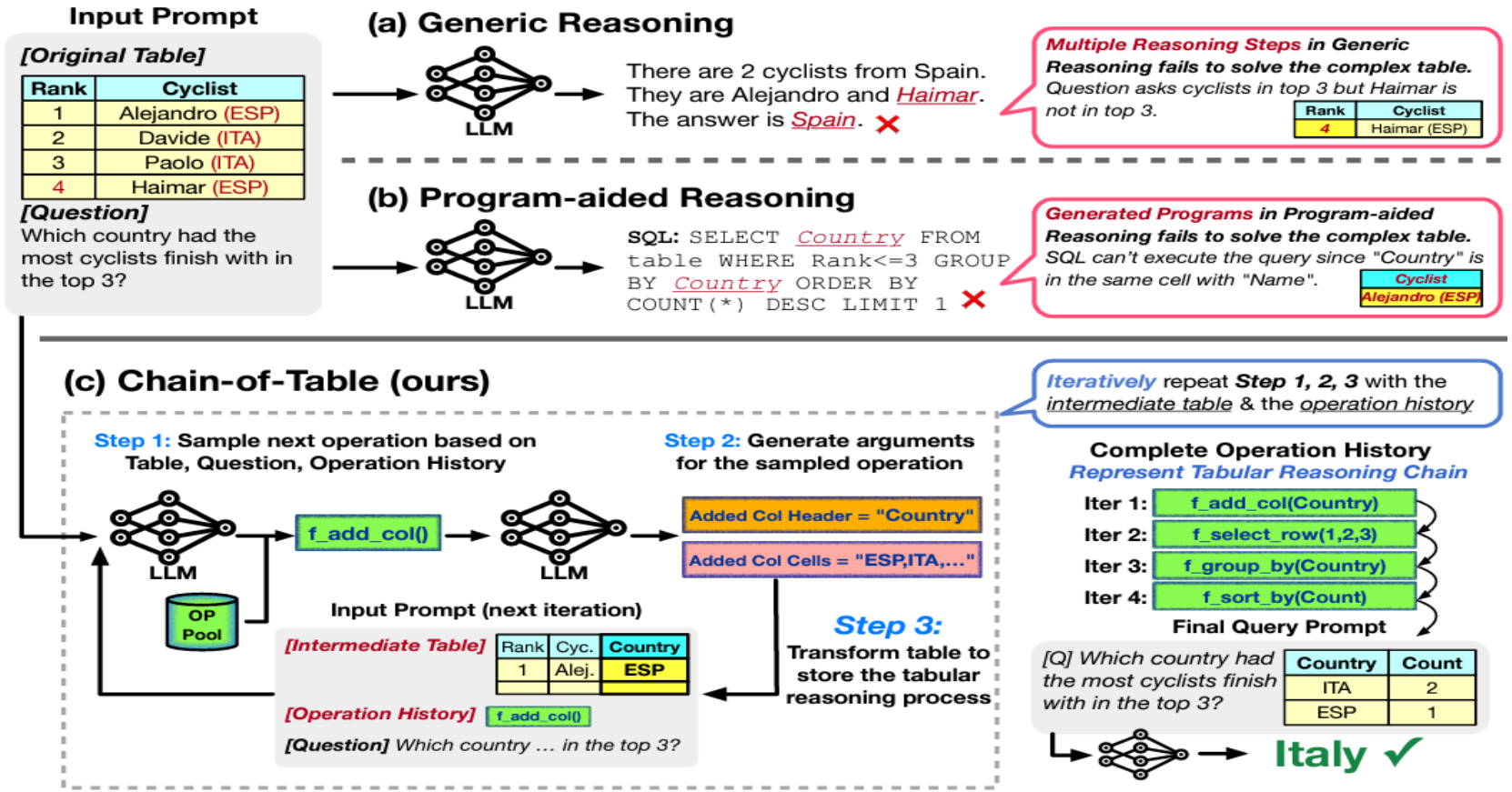



Figure 2: An overview of the proposed methodology including all four modules

https://arxiv.org/pdf/2304.11015

# Planning:

- **Task decomposition & Task selection & Reflection**
  - Chain-of-table [ICLR24]: TableQA, TabFact
  - Task decomposition: select action and generate arguments
  - Plan selection: DynamicPlan to select next action based on the current state




https://arxiv.org/pdf/2401.04398

# Action & Tool use:

- Action Space Design

| | Task | Action Space |
|---|---|---|
| Chain-of-Table | TableQA | Table actions:<br>[add_col, select_row, select_col, group_by, sort_by] |
| ToolQA | TableQA | Tools:<br>[text, database, math, graph, code, system] |
| SheetCopilot | Table Transformation | Virtual APIS of 40+ atomic actions<br>Categories: [manipulation, management, formatting, Charts. Pivot Table, Formula] |

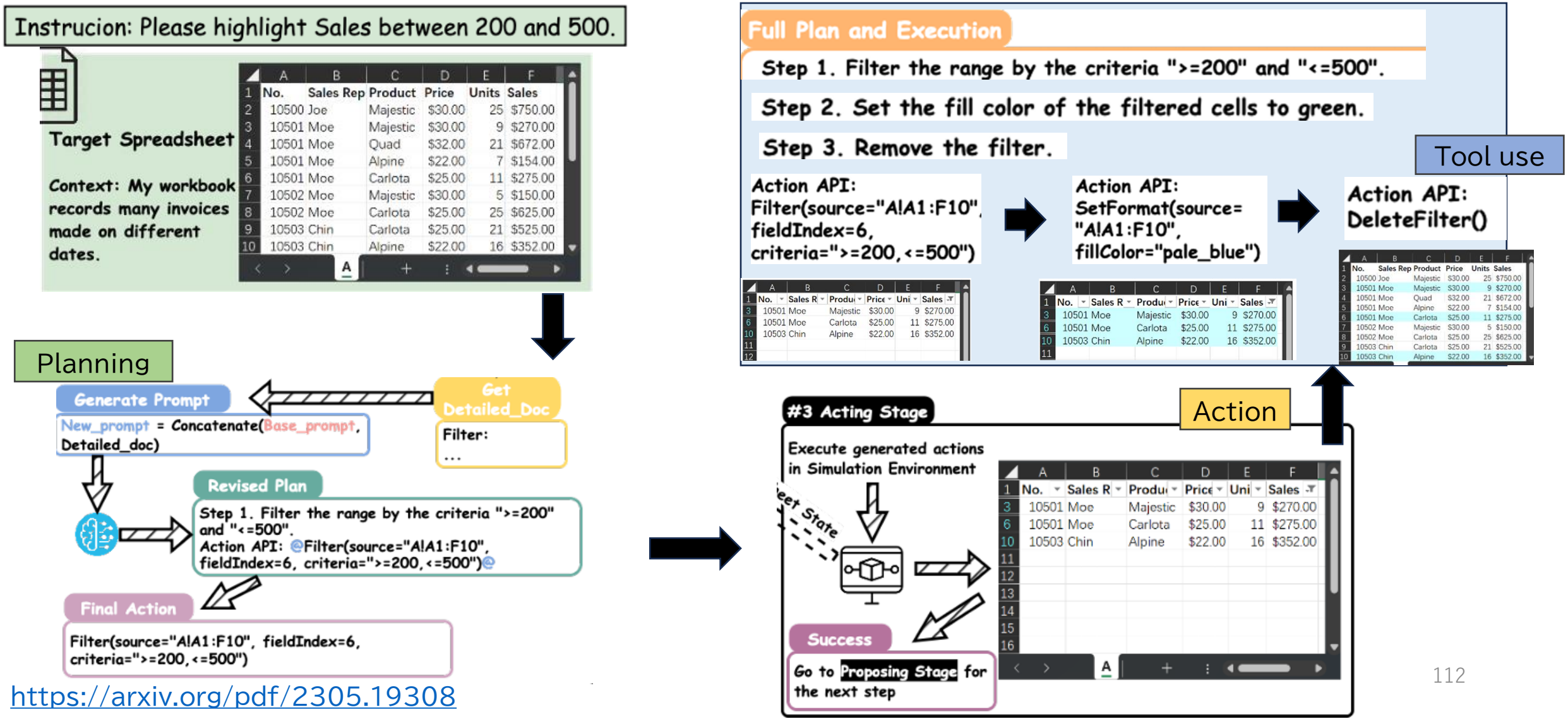
# Action & Tool use:

- External tool use

| | Task | SQL | Python | Retriever | Database | Math | Graph |
|---|---|---|---|---|---|---|---|
| PyAgent | TableQA | | ◯ | | | | |
| ToolQA | TableQA | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| ReAcTable | TableQA | ◯ | ◯ | | | | |
| LEVER | TableQA | ◯ | | | | | |
| Binder | TableQA, TableFact | ◯ | ◯ | ◯ | | | |
| Chain-of-Table | TableQA, TableFact | | ◯ | | | | |
| ToolWriter | TableQA | ◯ | ◯ | | | | |
| SheetAgent | Table Transformation | ◯ | ◯ | ◯ | | | |
| AutoTQA | TableQA, TableFact | ◯ | | ◯ | ◯ | | |
| TableRAG | TableQA, TableFact | | ◯ | ◯ | | | |

# Action & Tool use:

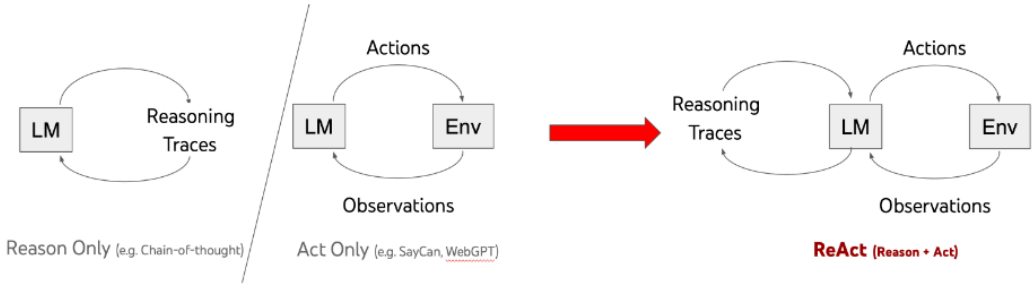
- SheetCopilot: planning, action & tool use (generate API call) for table transformation.




https://arxiv.org/pdf/2305.19308

# Action & Tool use:

- ReAcTable [VLDB24]
  - Apply ReAct [ICLR24], reason and action concept to table QA task
  - During action, use SQL and python shell



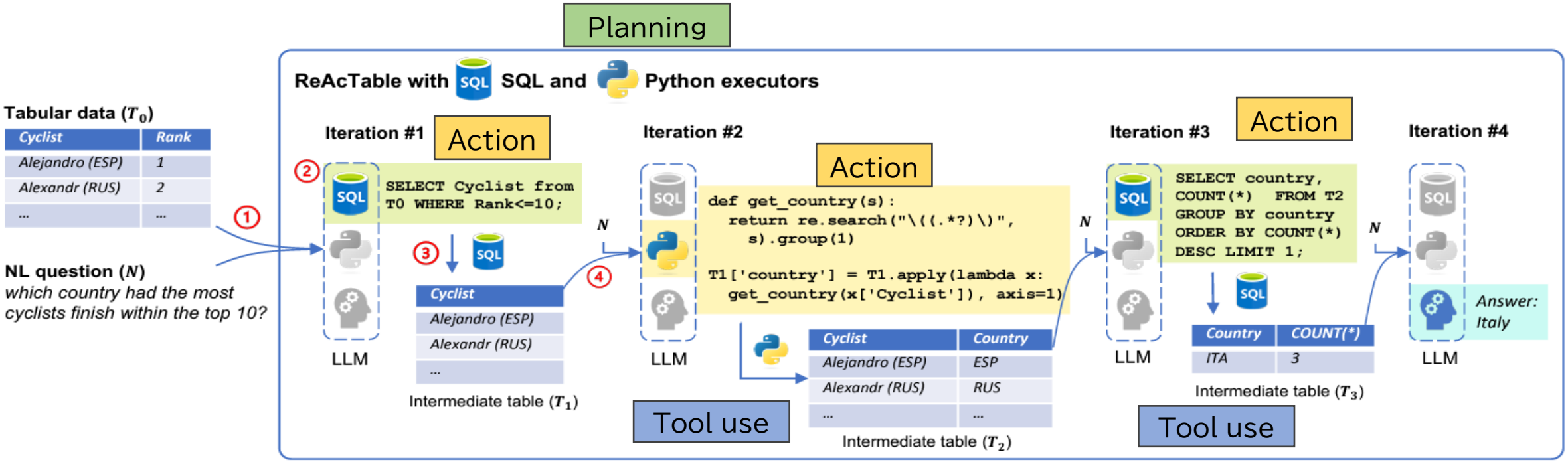
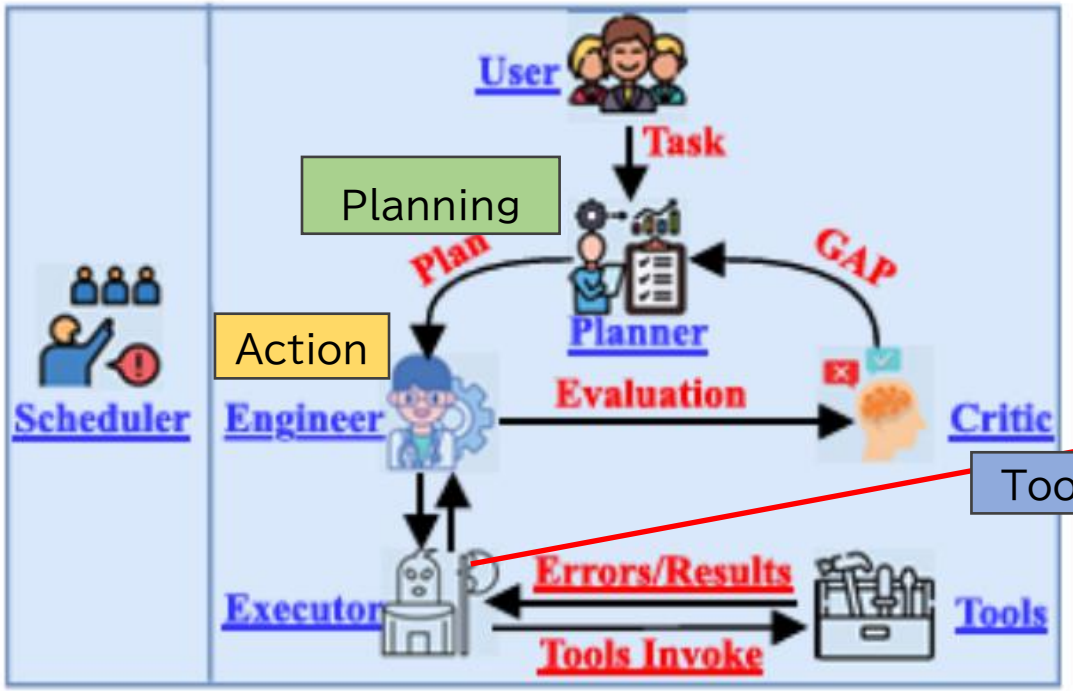




**Figure 1: Overview of the ReAcTable framework with SQL and Python code executors.**

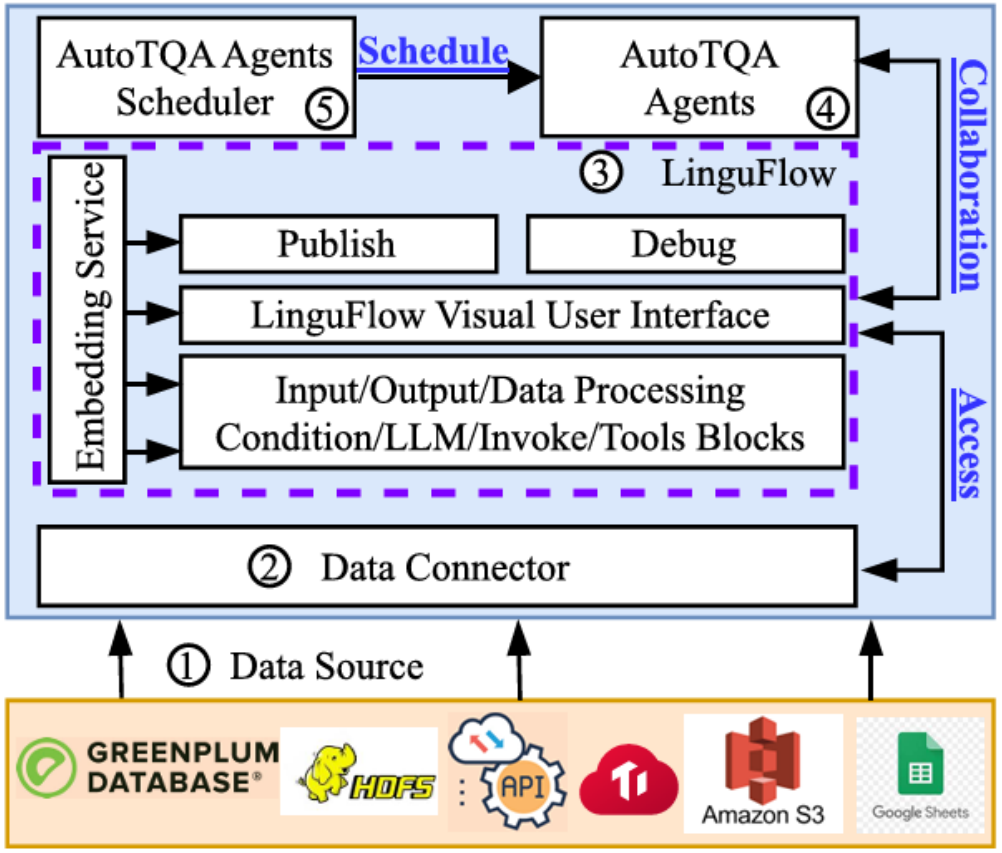
https://www.vldb.org/pvldb/vol17/p1981-zhang.pdf

# Action & Tool use:

- AutoTQA [VLDB24]: Use profile and separate action and tool use for agent cooperation.




Tool use




Profiles (Planner、Engineer、Executor and Critic)

https://www.vldb.org/pvldb/vol17/p3920-zhu.pdf

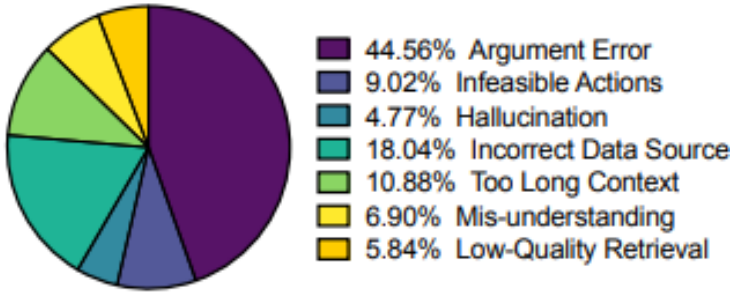
# Action & Tool use:

- ToolQA: a dataset to eval the ability of LLM use tool for table [arxiv23]
  - Include 13 tools in 8 domain

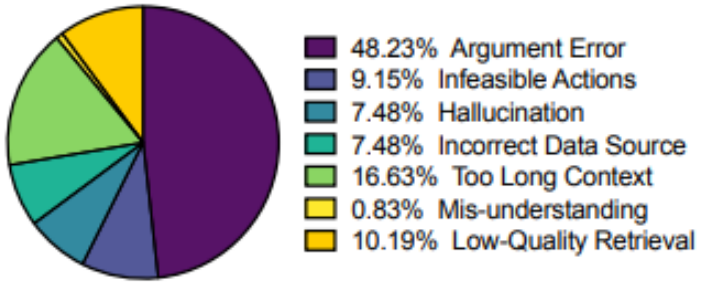
Table 2: Different tools in ToolQA.

| Tool Types | # Tools | Tools |
|---|---|---|
| Text Tools | 2 | Agenda Retriever, SciREX Retriever |
| Database Tools | 3 | Database Loader, Data Filter, Get Value |
| Math Tools | 1 | WolframAlpha Calculator |
| Graph Tools | 4 | Graph Loader, Neighbour Checker, Node Checker, Edge Checker |
| Code Tools | 2 | Python Interpreter, SQL Interpreter |
| System Tools | 1 | Finish |

| Error Type | Description |
|---|---|
| **Argument Error** | Incorrect parameters when calling tools |
| **Incorrect Data Source** | Difficulty in identifying the proper reference corpora |
| **Innovation and Hallucination** | Creative tool combinations leading to call a non-existent tool |
| **Infeasible Actions** | Attempting to use non-existent tools |
| **Too Long Context** | Exceeding model's length limitation |
| **Misunderstanding** | Inability to comprehend tool execution feedback |
| **Low-Quality Retrieval** | Insufficient relevant information extracted from corpora |




(a) Easy questions.




(b) Hard questions.

https://arxiv.org/abs/2306.13304

# Action & Tool use: Example of let LLM use program

You are working with a pandas dataframe in Python. The name of the dataframe is `df`. Your task is to use `python_repl_ast` to answer the question posed to you.

Profile

Tool description:
- `python_repl_ast`: A Python shell. Use this to execute python commands. Input should be a valid python command. When using this tool, …

Tool use

Guidelines:
- **Aggregated Rows**: Be cautious of rows that aggregate data such as 'total', 'sum', or 'average'. Ensure these rows do not influence your results inappropriately.
- **Data Verification**: Before concluding the final answer, always verify that your observations align with the original table and question.

Planning

Strictly follow the given format to respond:

Question: the input question you must answer
Thought: you should always think about what to do to interact with `python_repl_ast`
Action: can **ONLY** be `python_repl_ast`
Action Input: the input code to the action
Observation: the result of the action

Action

... (this Thought/Action/Action Input/Observation can repeat N times)
Thought: after verifying the table, observations, and the question, I am confident in the final answer
Final Answer: the final answer to the original input question (AnswerName1, AnswerName2...)

Notes for final answer:
- Ensure the final answer format is only "Final Answer: AnswerName1, AnswerName2..." form, no other form.
- Ensure the final answer is a number or entity names, as short as possible, without any explanation.
- Ensure to have a concluding thought that verifies the table, observations and the question before giving the final answer.

You are provided with a table regarding "[TITLE]". This is the result of `print(df.to_markdown())`:

[TABLE]

**Note**: All cells in the table should be considered as `object` data type, regardless of their appearance.

Begin!
Question: [QUESTION]
"""

# Action & Tool use: Example of let LLM use APIs

Profile

You are now capable of calling a weather API to retrieve weather information.

The API format is:POST /weather{ "city": "{city_name}"} Tool use

When I ask for the weather in a city, respond by generating the API call in this format.

Let's see a few examples:

User: What's the weather like in Tokyo?
Response: API Call: POST /weather{ "city": "Tokyo"}

User: Tell me the weather in New York.
Response: API Call: POST /weather{ "city": "New York"}

User: Can you check the weather in London?
Response: API Call: POST /weather{ "city": "London"}

Now it's your turn to generate the correct API call when I ask about a city's weather.

# Q&A

\Orchestrating a brighter world

NEC

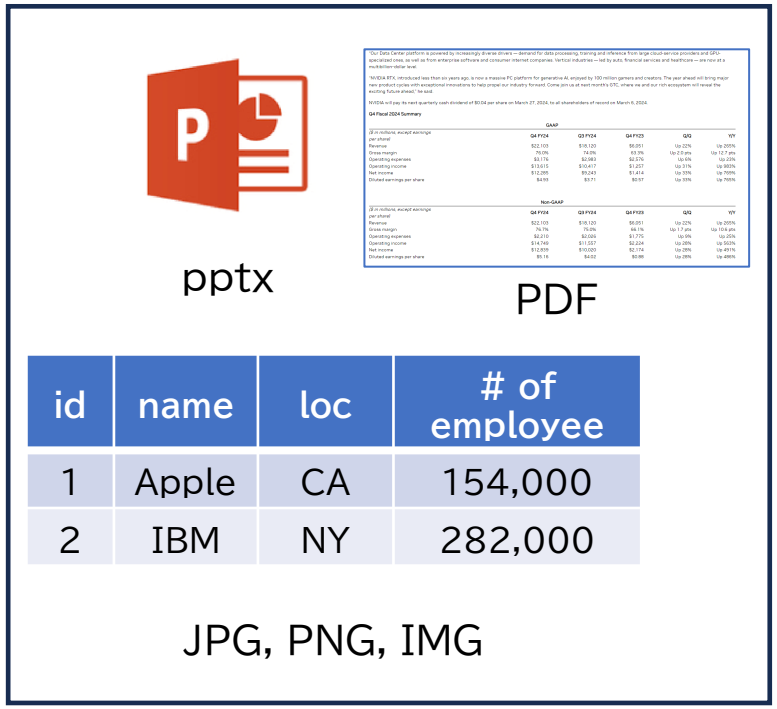
大阪大学
OSAKA UNIVERSITY

# On the Use of Large Language Models for Table Tasks

- Vision-language models (VLMs)

# What is VLM

- Input <prompt, image>, Output: response in text




VLM

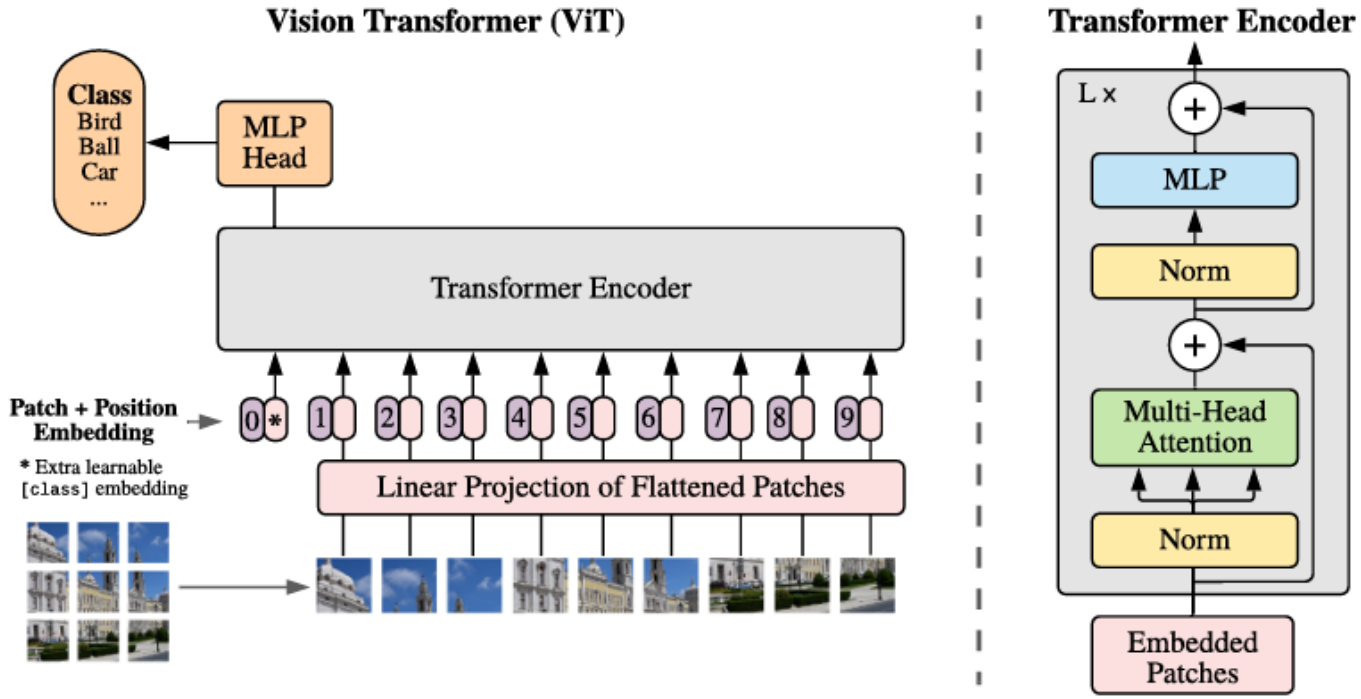
Response

The number of employees for Apple, as shown in the table, is 154,000.

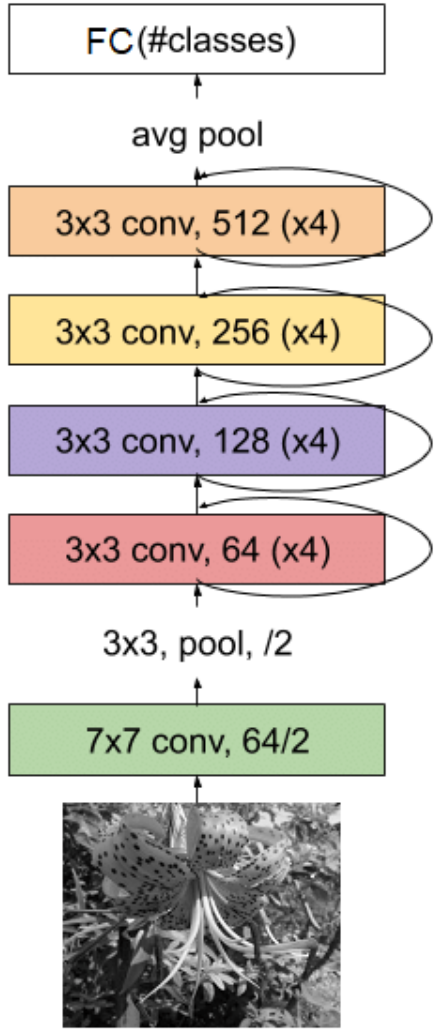
Prompt

what is the employee number of Apple?

# VIT (Vision Transformer) vs Resnet (CNN)



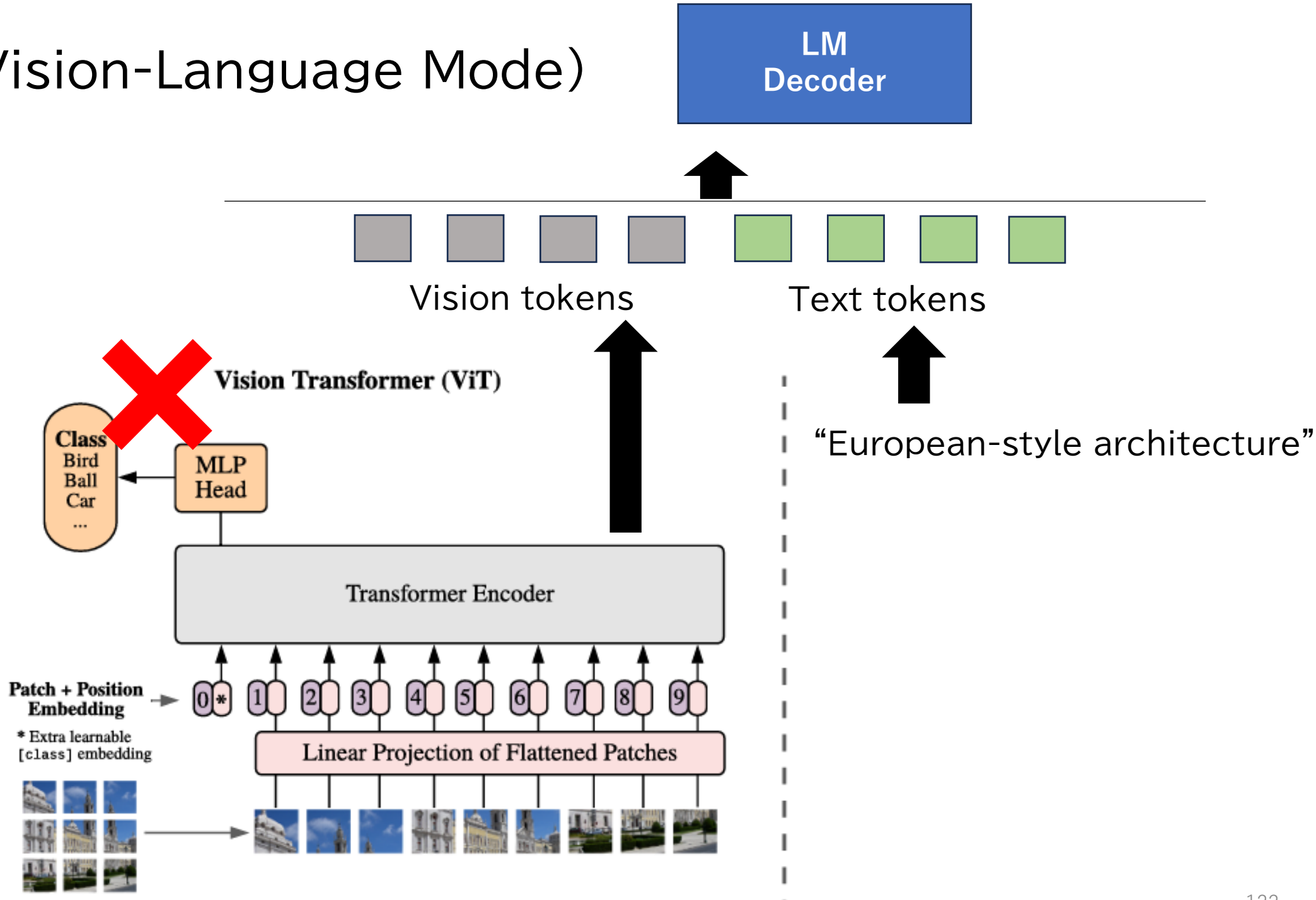

VIT




CNN

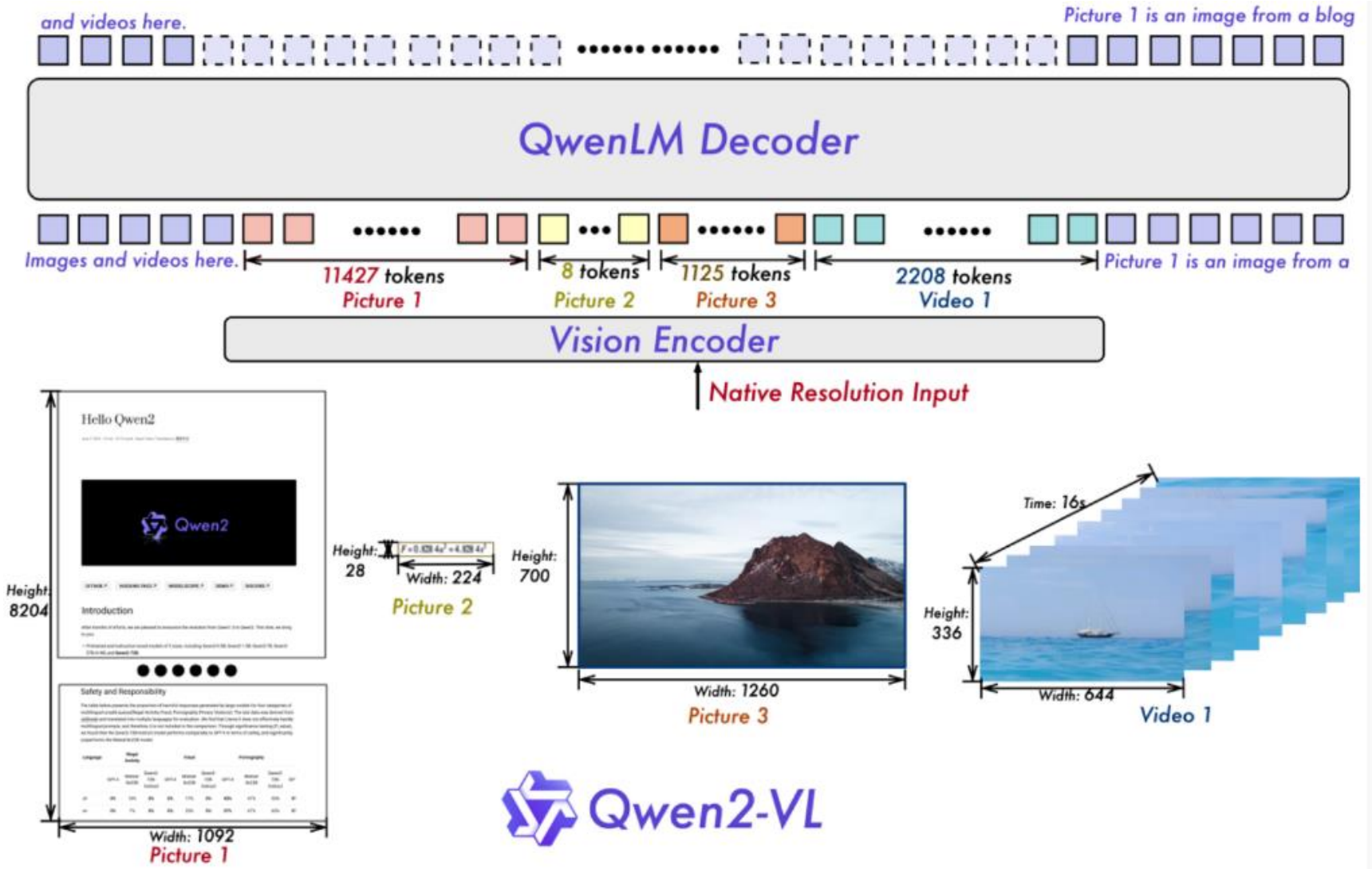
Vision Transformer [ICLR21] https://arxiv.org/abs/2010.11929

# VLM (Vision-Language Mode)



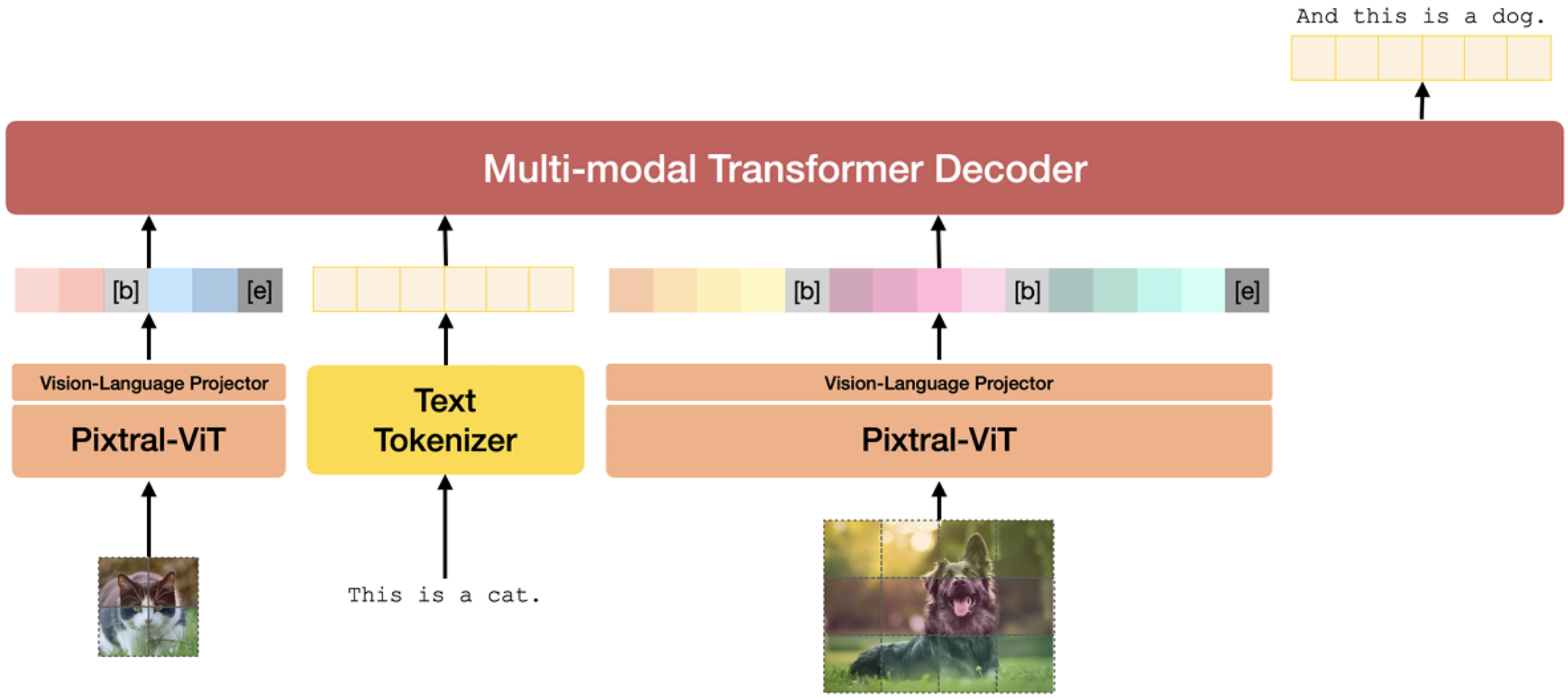

"European-style architecture"

# Qwen2-VL




https://qwen2.org/vl/

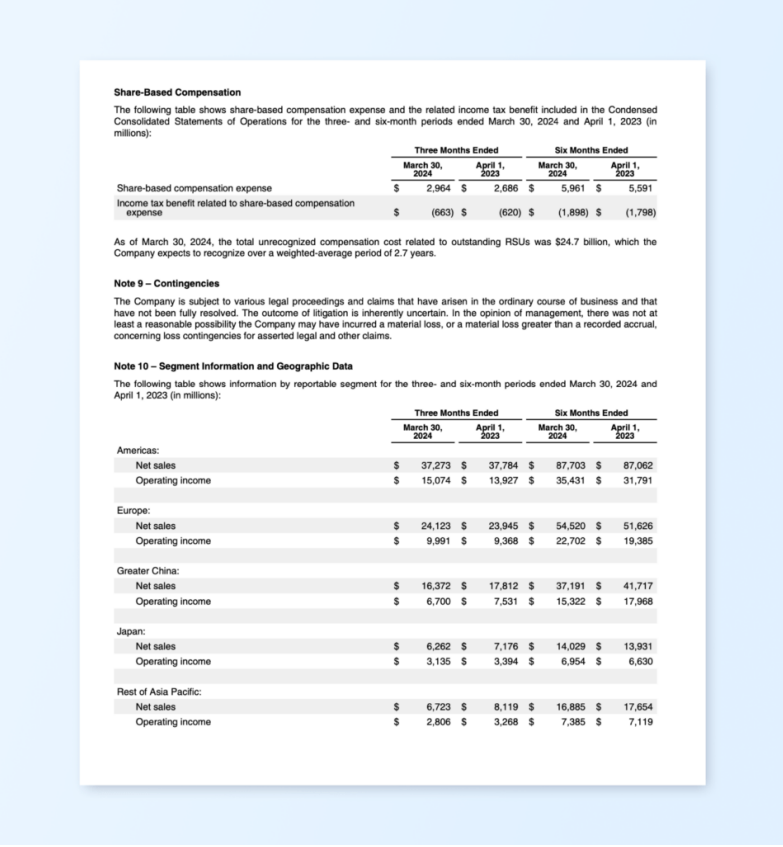
# Pixtral




https://arxiv.org/pdf/2410.07073

# Motivation of VLM for table

- Many real tables
  - Can not represent as text: markdown, JSON, HTML, latex, etc…
  - Understand with the text and vision features picture, marks, symbols in the document
- Understand table in a vision way is a natural choice



**Share-Based Compensation**

The following table shows share-based compensation expense and the related income tax benefit included in the Condensed Consolidated Statements of Operations for the three- and six-month periods ended March 30, 2024 and April 1, 2023 (in millions):

| | Three Months Ended | | Six Months Ended | |
|---|---|---|---|---|
| | March 30, 2024 | April 1, 2023 | March 30, 2024 | April 1, 2023 |
| Share-based compensation expense | $ 2,964 | $ 2,686 | $ 5,961 | $ 5,591 |
| Income tax benefit related to share-based compensation expense | $ (663) | $ (620) | $ (1,898) | $ (1,798) |

As of March 30, 2024, the total unrecognized compensation cost related to outstanding RSUs was $24.7 billion, which the Company expects to recognize over a weighted-average period of 2.7 years.

**Note 9 – Contingencies**

The Company is subject to various legal proceedings and claims that have arisen in the ordinary course of business and that have not been fully resolved. The outcome of litigation is inherently uncertain. In the opinion of management, there was not at least a reasonable possibility the Company may have incurred a material loss, or a material loss greater than a recorded accrual, concerning loss contingencies for asserted legal and other claims.

**Note 10 – Segment Information and Geographic Data**

The following table shows information by reportable segment for the three- and six-month periods ended March 30, 2024 and April 1, 2023 (in millions):

| | Three Months Ended | | Six Months Ended | |
|---|---|---|---|---|
| | March 30, 2024 | April 1, 2023 | March 30, 2024 | April 1, 2023 |
| Americas: | | | | |
| Net sales | $ 37,273 | $ 37,784 | $ 87,703 | $ 87,062 |
| Operating income | $ 15,074 | $ 13,927 | $ 35,431 | $ 31,791 |
| Europe: | | | | |
| Net sales | $ 24,123 | $ 23,945 | $ 54,520 | $ 51,626 |
| Operating income | $ 9,991 | $ 9,368 | $ 22,702 | $ 19,385 |
| Greater China: | | | | |
| Net sales | $ 16,372 | $ 17,812 | $ 37,191 | $ 41,717 |
| Operating income | $ 6,700 | $ 7,531 | $ 15,322 | $ 17,968 |
| Japan: | | | | |
| Net sales | $ 6,262 | $ 7,176 | $ 14,029 | $ 13,931 |
| Operating income | $ 3,135 | $ 3,394 | $ 6,954 | $ 6,630 |
| Rest of Asia Pacific: | | | | |
| Net sales | $ 6,723 | $ 8,119 | $ 16,885 | $ 17,654 |
| Operating income | $ 2,806 | $ 3,268 | $ 7,385 | $ 7,119 |

**FEATURE COMPARISON**

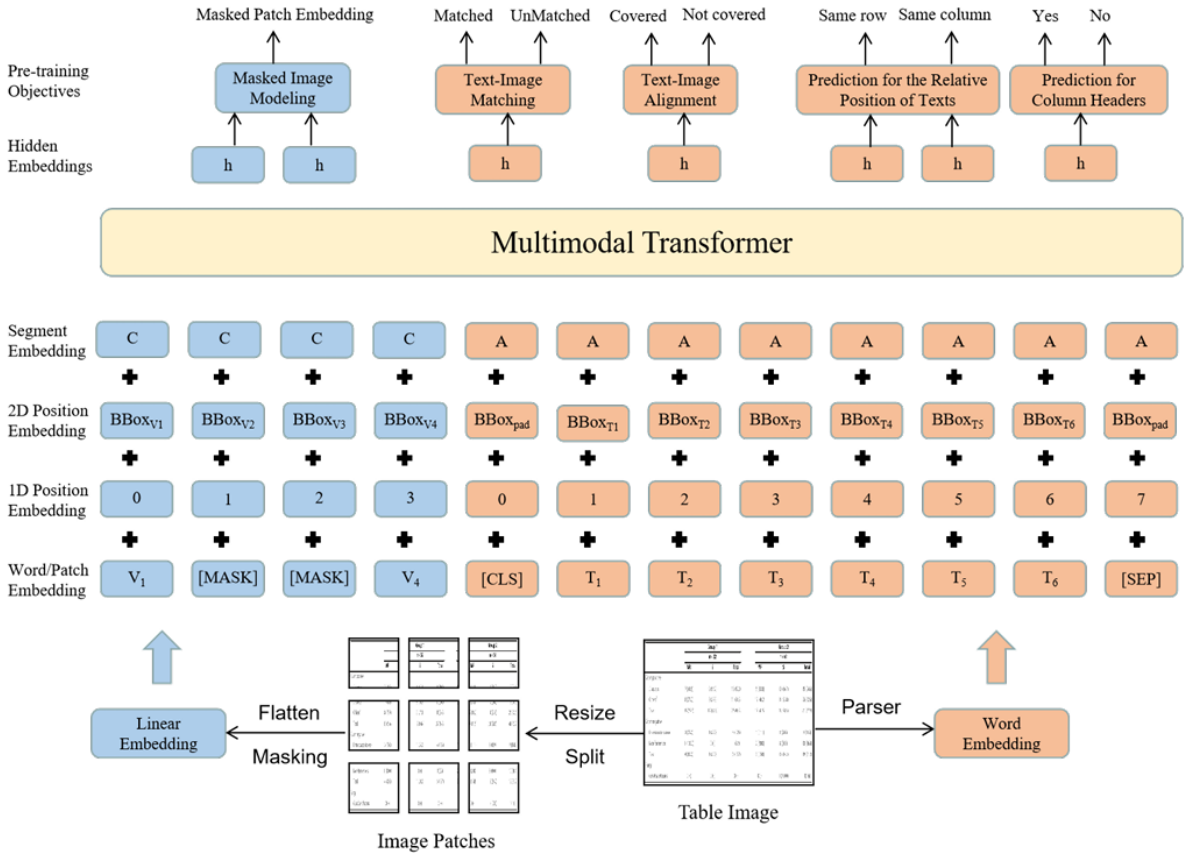
vertex42



| FEATURES | PRODUCT NAME 1 | PRODUCT NAME 2 | PRODUCT NAME 3 | PRODUCT NAME 4 | PRODUCT NAME 5 | PRODUCT NAME 6 |
|---|---|---|---|---|---|---|
| PRICE | FREE | $25-$30 | $45-$60 | $199 | FREE | $500 |
| Unicode ✓ or × symbols | ✓ | × | ✓ | | × | ✓ |
| Custom Icon Sets | ○ | ✓ | ✗ | ✓ | | ✗ |
| Up / Down Arrow Icons | ▼ | ▬ | ▲ | | ▼ | ▲ |
| Check / X Icons | ✗ | ! | ✓ | ✓ | | ✓ |
| Flag Icons | ⚑ | ⚑ | ⚑ | ⚑ | ⚑ | |
| Ratings | | | | | | |
| Unicode ★ symbol | ★★★★ | ★ | ★★ | ★★★ | | ★★★★★ |
| Diamond ◆ symbol | ◆ | ◆◆ | ◆◆◆ | ◆◆ | ◆ | ◆◆◆ |
| Icon Set: Star (0,1,2) | ☆ | ☆ | ☆ | ☆ | | ☆ |
| Icon Set: Quarter Circle (0,1,2,3,4) | ○ | ◔ | ◑ | ◕ | ○ | ● |
| Icon Set: Boxes (0,1,2,3,4) | ▯ | ▯ | ▯ | ▯ | | ▯ |
| Icon Set: Bars (0,1,2,3,4) | ▁ | ▂ | ▃ | ▄ | ▁ | ▅ |
| Numeric & Text Specifications | | | | | | |
| Data Bars | 30 | 25 | 15 | 25 | 5 | 90 |
| Custom number formats | 128 GB | 64 GB | 256 GB | 512 GB | 32 GB | 1024 GB |
| Basic Numeric Entry | 4.23 | 1.23 | 5 | 6.2 | 1.54 | 3.52 |
| Yes / No / na | No | Yes | No | Yes | na | Yes |
| Numbers with Different Units | 2.3 TB | 470 GB | 125 GB | 64 GB | 512 MB | 128 GB |

# TableVLM(ACL23)

- Table detection and structure recognition task
  - Input: table image
  - Output: table html
- Encoder Pretrain tasks (12 layers)
  - Text-Image alignment
    - Predict if an image contain a text
  - Text-Image matching
    - Predict if image and text matched
  - Masked image modeling
- Decoder Pretrain task (4 layers)
  - Generate HTML




https://aclanthology.org/2023.acl-long.137.pdf

# TableVLM(ACL23)

- Datasets and experimental results

| Datasets | Source | Format | Sizes |
|---|---|---|---|
| Marmot | e-Books and Citeseer website | bmp, xml | 958 |
| ICDAR 2013 | European Union and US Government websites | pdf, xml | 150 |
| ICDAR 2019 | modern and archival documents with various formats | jpg, xml | 3.6k |
| TableBank | Word and Latex documents on the internet | jpg, HTML | 145k |
| SciTSR | LaTeX source files | pdf, Latex | 15k |
| PubTabNet | scientific articles in PMCOA | png, HTML | 568k |
| TabLeX | scientific paper from arXiv | jpg, Latex | 3, 00k |
| FinTabNet | annual reports of the S&P 500 companies | png, HTML | 112k |
| SynthTabNet | synthetically generated based on Tablebank, PubTabNet, and FinTabNet | png, HTML | 600k |
| ComplexTable (ours) | synthetically generated by an auto HTML table creator | png, HTML | 1, 000k |

Table 1: Existing public datasets available and the constructed ComplexTable dataset for table structure recognition.

| Model | Dataset | Simple | Complex | All |
|---|---|---|---|---|
| WYGIWS | TableBank | 86.4 | −− | 86.4 |
| EDD | TableBank | 86.0 | −− | 86.0 |
| LGPMA | TableBank | 88.7 | −− | 88.7 |
| Master | TableBank | 89.4 | −− | 89.4 |
| TableFormer | TableBank | 89.6 | −− | 89.6 |
| TableVLM | TableBank | **90.2** | −− | **90.2** |
| LGPMA | PubTabNet | 97.88 | 94.78 | 96.36 |
| Master | PubTabNet | 97.90 | 94.68 | 96.32 |
| TableFormer | PubTabNet | **98.5** | 95.0 | 96.8 |
| TableVLM | PubTabNet | 98.31 | **95.53** | **96.92** |
| LGPMA | ComplexTable | 90.54 | 86.87 | 88.76 |
| Master | ComplexTable | 92.17 | 88.79 | 90.21 |
| TableVLM | ComplexTable | **94.73** | **90.43** | **92.18** |

Table 2: The tree-edit-distance-based similarity (TEDS) of table structure recognition on TableBank, PubTabNet and ComplexTable datasets. A table is categorized as a simple table if it lacks multi-column or multi-row cells; otherwise, it is classified as a complex table. It is worth noting that the TableBank dataset does not include any complex tables.



https://aclanthology.org/2023.acl-long.137.pdf

# PixT3 (ACL24)

- Table to text task
  - Input: table image
  - Output: text description of table
- Encoder-decoder Pretrain task
  - Generate row and column with a cell
- Finetune different models on specific table to text tasks

Table:

| | | | |
|---|---|---|---|
| oY | io | HG | eG2S |
| Z4ikU | O1 | aRU | mubk6 |
| URa | dAF | | I |
| I86 | GAe | Ob | sUr5 |
| L1 | 3 | Vf1 | Svaq2 |

Target:

<<<dAF><<<URa><I>>><<<io><O1><GAe>
<3>><<<HG><aRU><Ob><Vf1>>>>>

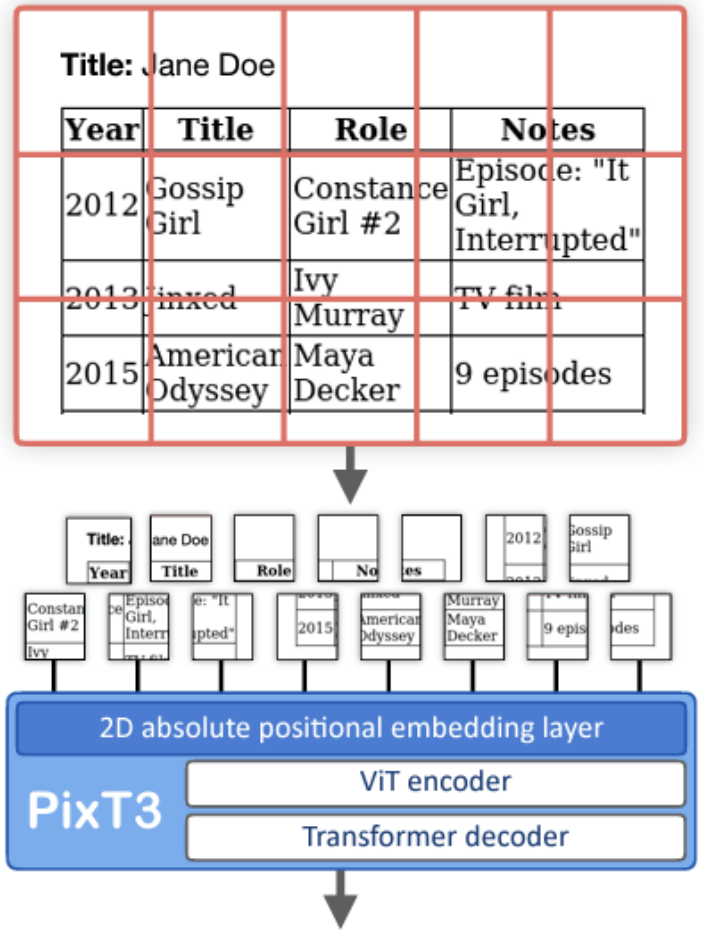
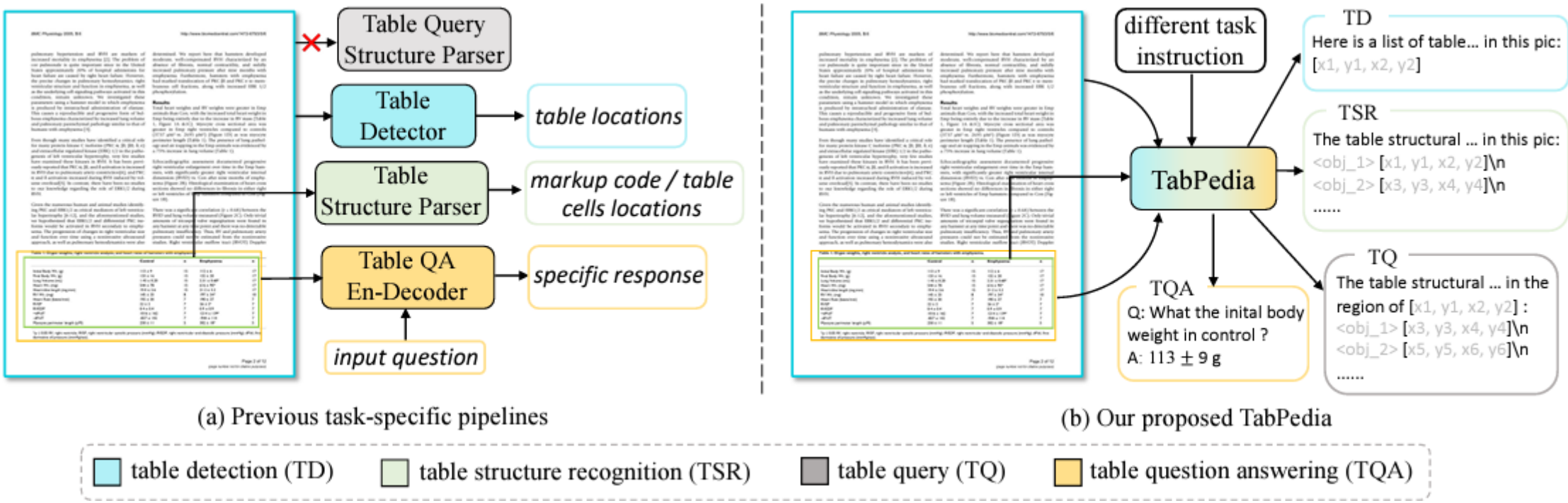



Figure 2: Overview of PixT3 generation model.

https://arxiv.org/abs/2311.09808

# Tabpedia (arXiv24)

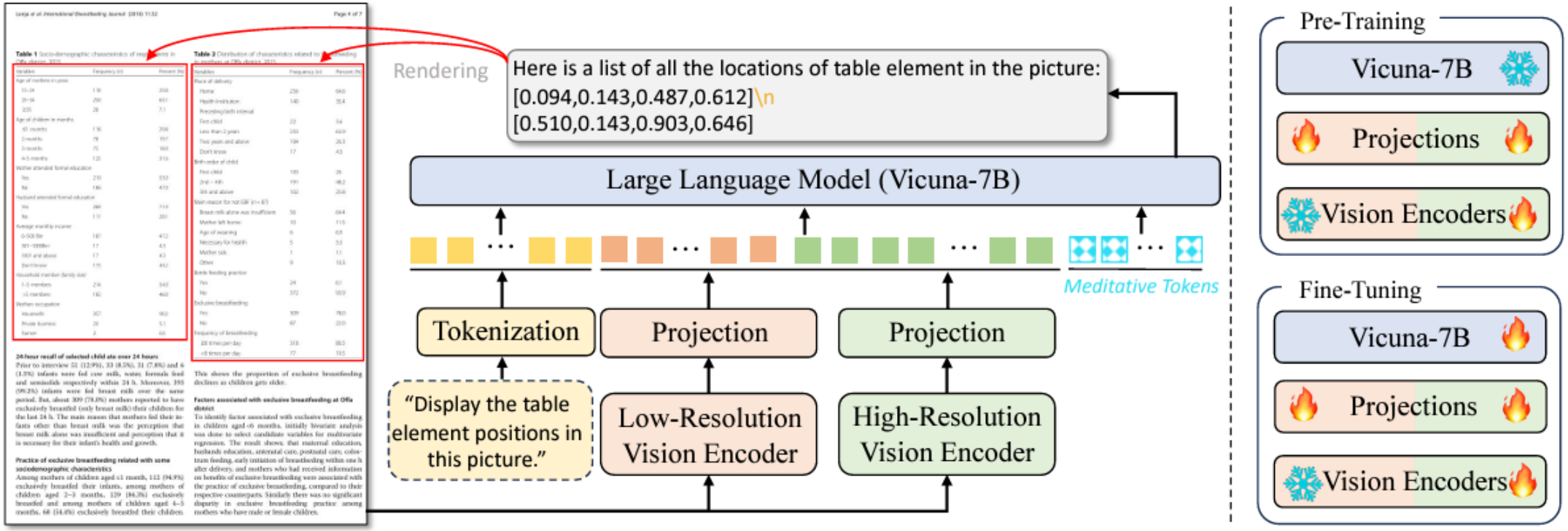
- multi-task: table detection& structure recognition, tableQA




https://arxiv.org/pdf/2406.01326

# Tabpedia (arXiv24)

- Architecture



# Tabpedia (arXiv24)

- Training data and tasks

Table 1: Summary of training data statistics in the fine-tuning stage.

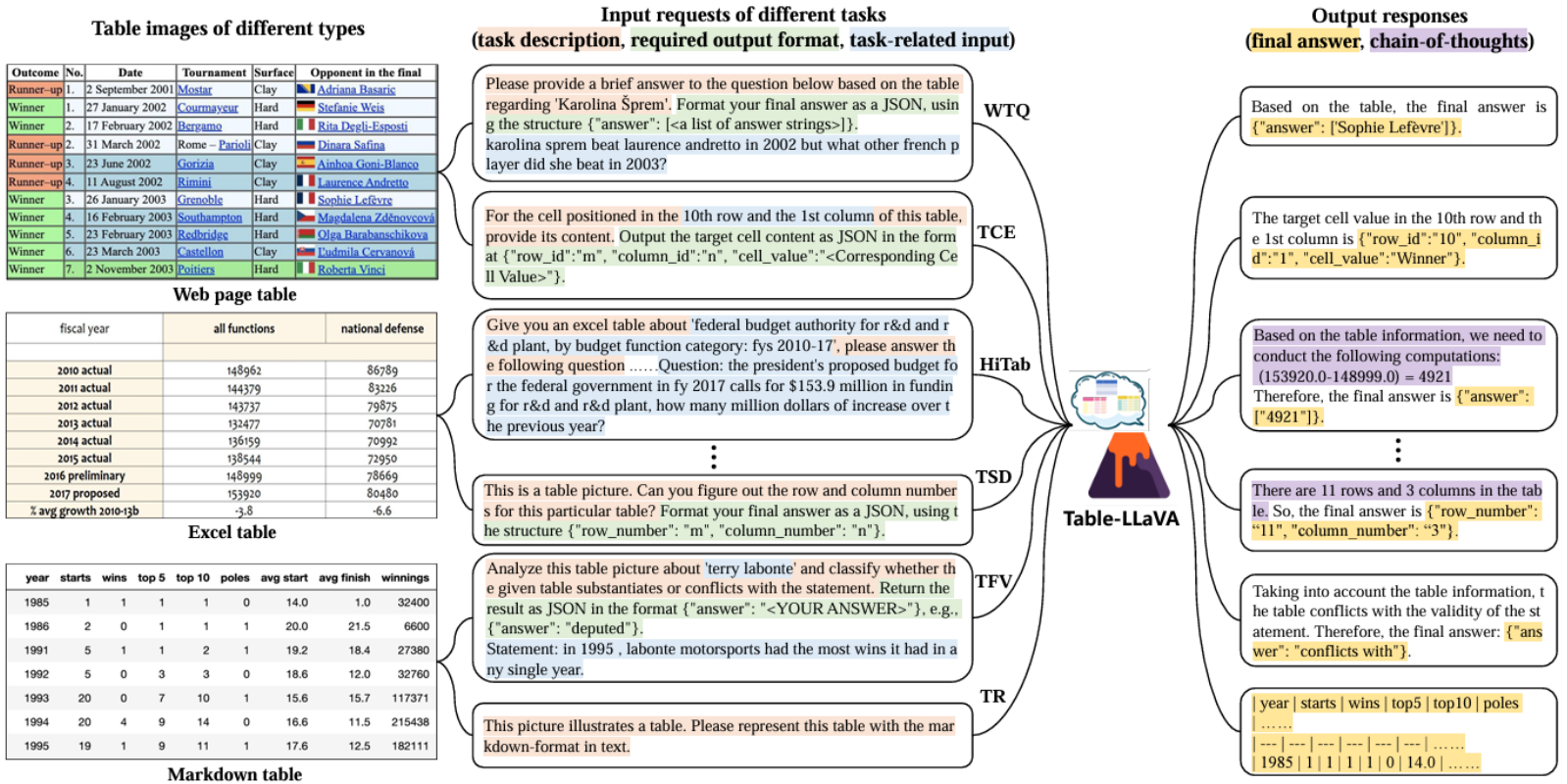
| Dataset | Subset | Task | Num |
|---|---|---|---|
| PubTab1M | PubTab1M-Det | TD | 460k |
| | PubTab1M-Str | TSR,TQA | 759k |
| | PubTab1M-Syn | TQ | 381k |
| FinTabNet | – | TSR,TQA | 78k |
| PubTabNet | – | TSR | 434k |
| WTQ | – | TQA | 1k |
| TabFact | – | TQA | 9k |

Table 2: Different task types and their instruction examples.

| Task | Example |
|---|---|
| TD | "Give me the areas where table element's locations in this picture." |
| TSR | "Parse the structural information of the cropped table in this picture." |
| TQ | "Parse the table structure within the region [0.095, 0.673, 0.869, 0.851] in this picture." |
| TQA | "What was the lowest stock price in the fourth quarter of 2010?" |

# Table-Llava (ACL24)

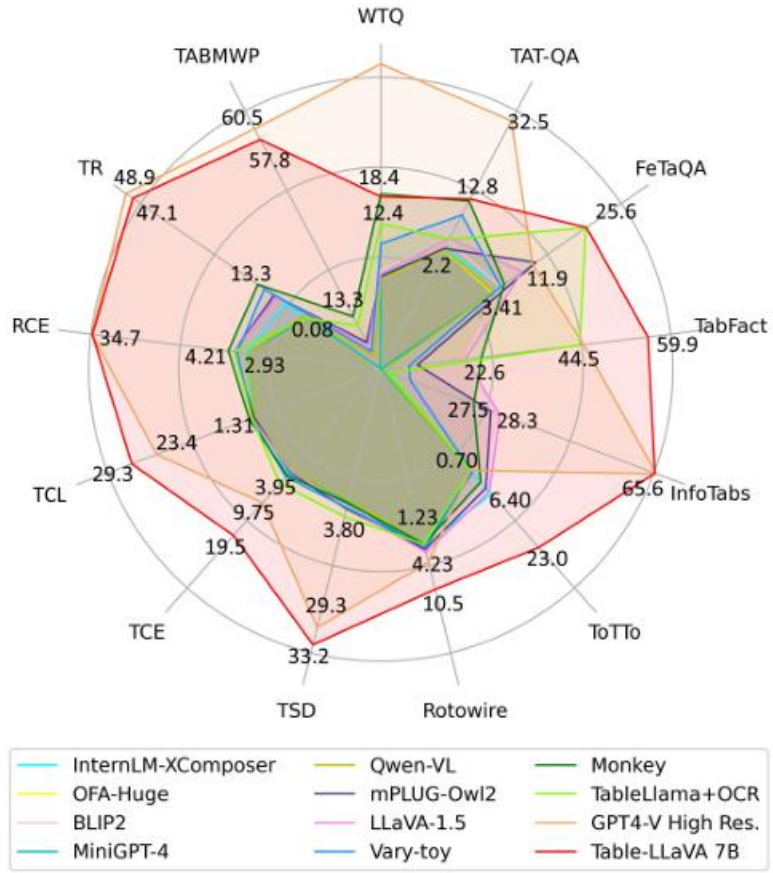
- A general VLM for multitasks




| Outcome | No. | Date | Tournament | Surface | Opponent in the final |
|---|---|---|---|---|---|
| Runner–up | 1. | 2 September 2001 | Mostar | Clay | Adriana Basaric |
| Winner | 1. | 27 January 2002 | Courmayeur | Hard | Stefanie Weis |
| Winner | 2. | 17 February 2002 | Bergamo | Hard | Rita Degli-Esposti |
| Runner–up | 2. | 31 March 2002 | Rome – Parioli | Clay | Dinara Safina |
| Runner–up | 3. | 23 June 2002 | Gorizia | Clay | Ainhoa Goni-Blanco |
| Runner–up | 4. | 11 August 2002 | Rimini | Clay | Laurence Andretto |
| Winner | 3. | 26 January 2003 | Grenoble | Hard | Sophie Lefèvre |
| Winner | 4. | 16 February 2003 | Southampton | Hard | Magdalena Zděnovcová |
| Winner | 5. | 23 February 2003 | Redbridge | Hard | Olga Barabanschikova |
| Winner | 6. | 23 March 2003 | Castellon | Clay | Ludmila Cervanová |
| Winner | 7. | 2 November 2003 | Poitiers | Hard | Roberta Vinci |

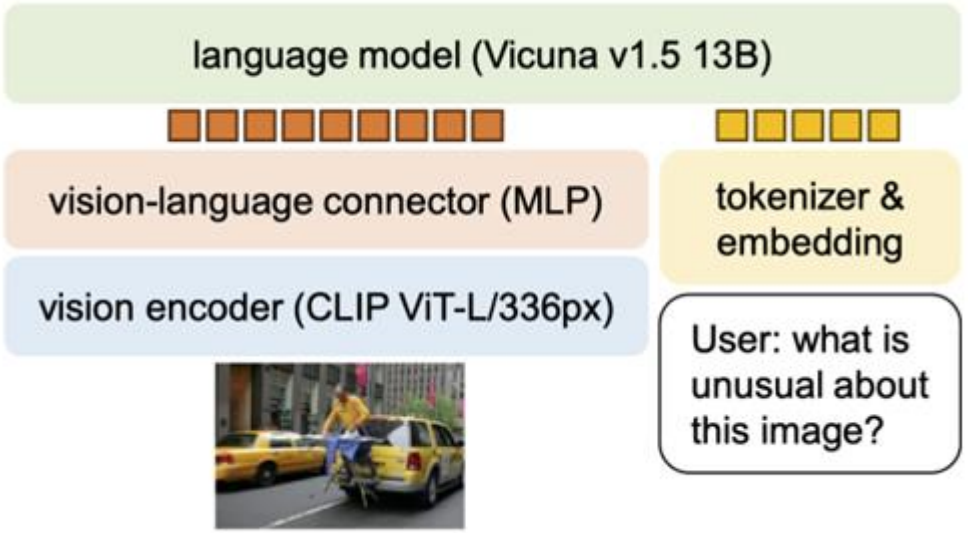
| fiscal year | all functions | national defense |
|---|---|---|
| 2010 actual | 148962 | 86789 |
| 2011 actual | 144379 | 83226 |
| 2012 actual | 143737 | 79875 |
| 2013 actual | 132477 | 70781 |
| 2014 actual | 136159 | 70992 |
| 2015 actual | 138544 | 72950 |
| 2016 preliminary | 148999 | 78669 |
| 2017 proposed | 153920 | 80480 |
| % avg growth 2010-13b | -3.8 | -6.6 |

| year | starts | wins | top 5 | top 10 | poles | avg start | avg finish | winnings |
|---|---|---|---|---|---|---|---|---|
| 1985 | 1 | 1 | 1 | 1 | 0 | 14.0 | 1.0 | 32400 |
| 1986 | 2 | 0 | 1 | 1 | 1 | 20.0 | 21.5 | 6600 |
| 1991 | 5 | 1 | 1 | 2 | 1 | 19.2 | 18.4 | 27380 |
| 1992 | 5 | 0 | 3 | 3 | 0 | 18.6 | 12.0 | 32760 |
| 1993 | 20 | 0 | 7 | 10 | 1 | 15.6 | 15.7 | 117371 |
| 1994 | 20 | 4 | 9 | 14 | 0 | 16.6 | 11.5 | 215438 |
| 1995 | 19 | 1 | 9 | 11 | 1 | 17.6 | 12.5 | 182111 |




https://arxiv.org/abs/2406.08100

# Table-Llava (ACL24)




Llava architecture

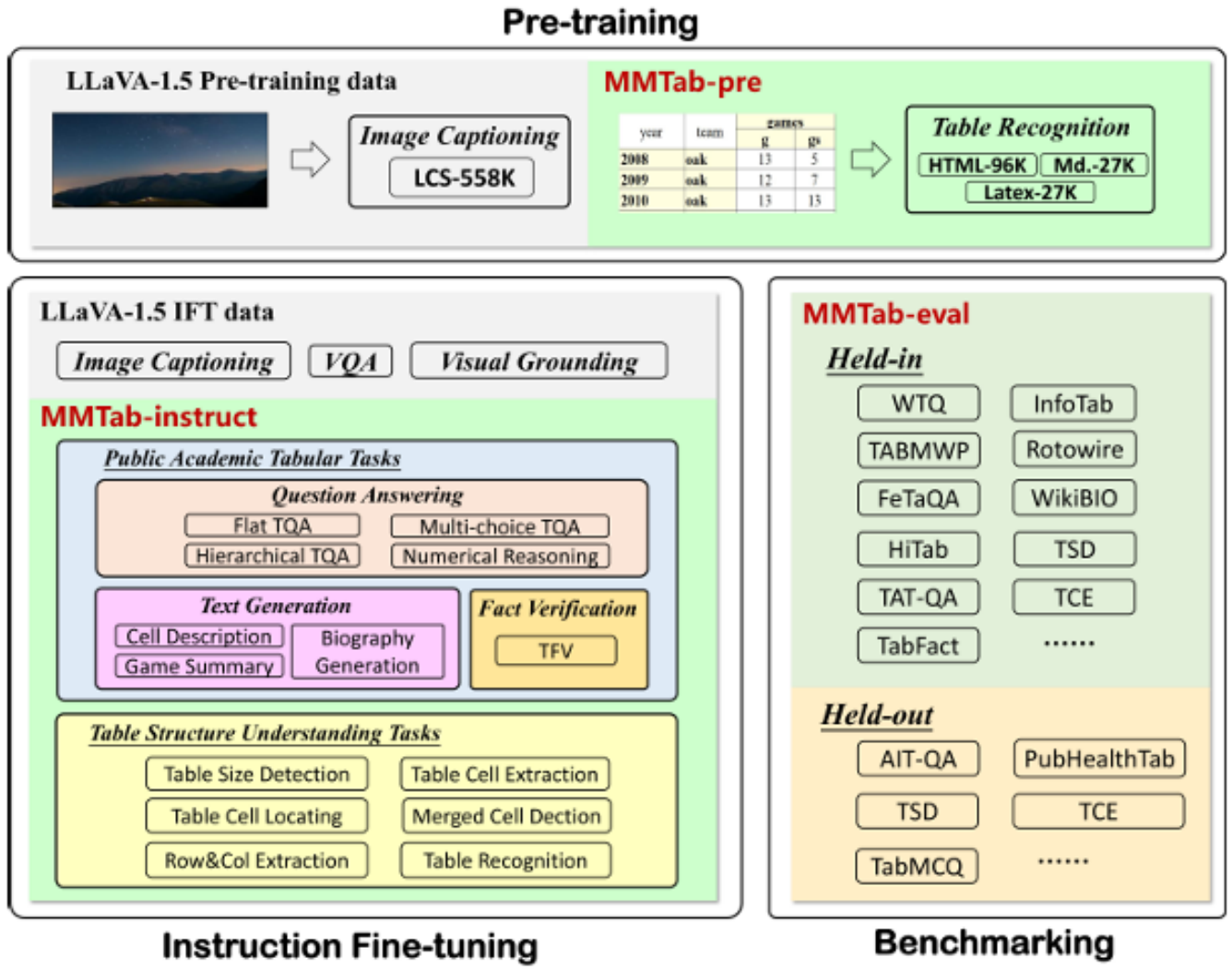


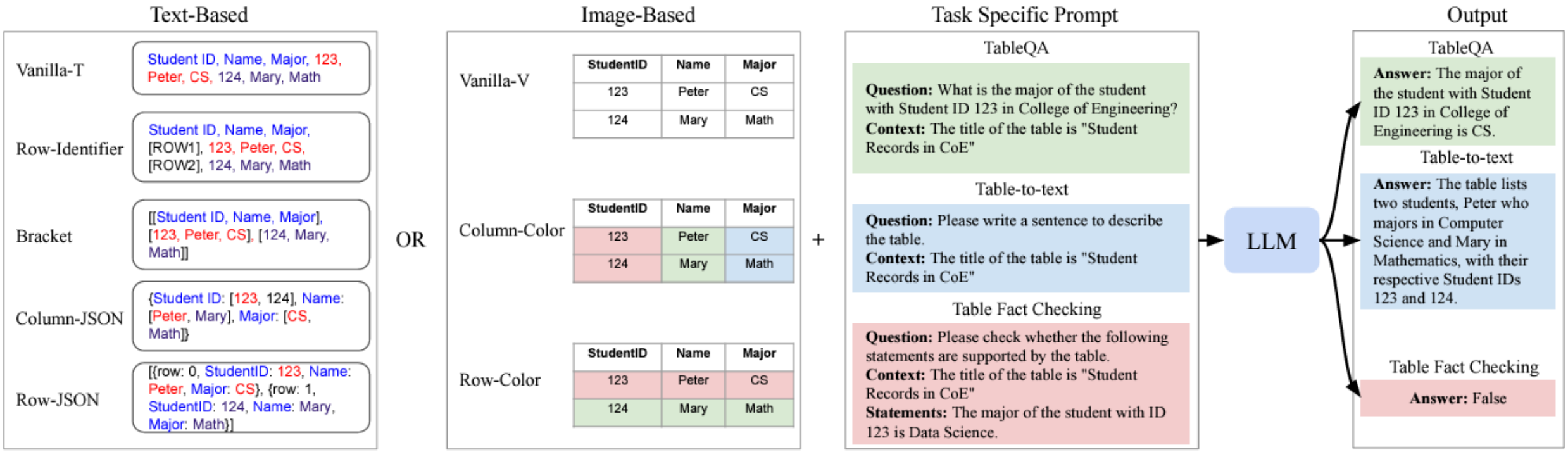

Table-Llava training

# Summary of current table VLMs

| | Vision encoder | LM decoder backbone | Task |
|---|---|---|---|
| Tableformer [CVPR22] | Resnet-18(CNN) + transformer encoder (2 layers) | Scratch transformer decoder (4 layers) | Table detection & structure recognition |
| TableVLM [ACL23] | ResNeXt101 (CNN) + transformer encoder (12 layers) | Scratch transformer decoder (4 layers) | Table detection & structure recognition |
| PixT3 [ACL24] | Pix2Strct (ViT) | Pix2Strct | Table-to-text |
| TablePedia [arxiv24] | ViT-L (ViT)<br>Swin-B (ViT) | Vicuna-7B | Table detection & structure recognition<br>TableQA |
| Table-Llava [ACL24] | CLIP (ViT) | Vicuna-v1.5-7B | Table-to-text<br>TableQA<br>Table structure recognition |

# Table as Texts or Images (ACL24)

- Evaluation the performance for different table formats.




https://arxiv.org/pdf/2402.12424

# Table as Texts or Images (ACL24)

| Method Name | Table Representation |
|---|---|
| Vanilla-T | $c_1, c_2, \cdots, c_n, v_{(1,1)}, v_{(1,2)}, \cdots, v_{(1,n)}, v_{(2,1)}, v_{(2,2)}, \cdots, v_{(2,n)}, \cdots, v_{(m,1)}, v_{(m,2)}, \cdots, v_{(m,n)}$. |
| Row-Identifier | $c_1, c_2, \cdots, c_n$, [ROW1] $v_{(1,1)}, v_{(1,2)}, \cdots, v_{(1,n)}$, [ROW2] $v_{(2,1)}, v_{(2,2)}, \cdots, v_{(2,n)}, \cdots$, [ROW$m$] $v_{(m,1)}, v_{(m,2)}, \cdots, v_{(m,n)}$. |
| Bracket | $[[c_1, c_2, \cdots, c_n], [v_{(1,1)}, v_{(1,2)}, \cdots, v_{(1,n)}], [v_{(2,1)}, v_{(2,2)}, \cdots, v_{(2,n)}], \cdots, [v_{(m,1)}, v_{(m,2)}, \cdots, v_{(m,n)}]]$. |
| Column-JSON | $\{ c_1: [v_{(1,1)}, v_{(2,1)}, \cdots, v_{(m,1)}], c_2: [v_{(1,2)}, v_{(2,2)}, \cdots, v_{(m,2)}], \cdots, c_n: [v_{(1,n)}, v_{(2,n)}, \cdots, v_{(m,n)}] \}$. |
| Row-JSON | [{ Row: 1, $c_1: v_{(1,1)}, c_2: v_{(1,2)}, \cdots, c_n: v_{(1,n)}$}, { Row: 2, $c_1: v_{(2,1)}, c_2: v_{(2,2)}, \cdots, c_n: v_{(2,n)}$ }, $\cdots$, { Row: m, $c_1: v_{(m,1)}, c_2: v_{(m,2)}, \cdots, c_n: v_{(m,n)}$ }]. |

Text representation

| $c_1$ | $c_2$ | … | $c_n$ |
|---|---|---|---|
| $v_{(1,1)}$ | $v_{(1,2)}$ | … | $v_{(1,n)}$ |
| $v_{(2,1)}$ | $v_{(2,2)}$ | … | $v_{(2,n)}$ |
| | | ⋮ | |
| $v_{(m,1)}$ | $v_{(m,2)}$ | … | $v_{(m,n)}$ |

Vanilla-V

| $c_1$ | $c_2$ | … | $c_n$ |
|---|---|---|---|
| $v_{(1,1)}$ | $v_{(1,2)}$ | … | $v_{(1,n)}$ |
| $v_{(2,1)}$ | $v_{(2,2)}$ | … | $v_{(2,n)}$ |
| | | ⋮ | |
| $v_{(m,1)}$ | $v_{(m,2)}$ | … | $v_{(m,n)}$ |

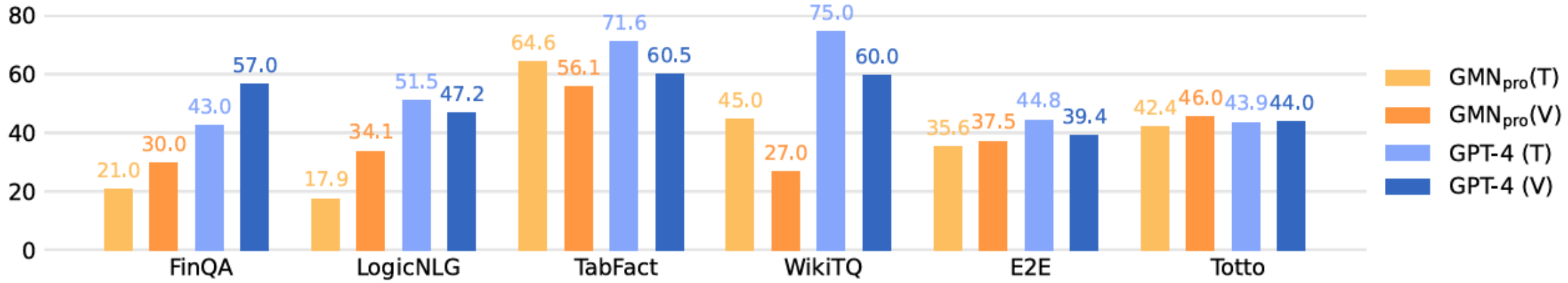
Column-Color

| $c_1$ | $c_2$ | … | $c_n$ |
|---|---|---|---|
| $v_{(1,1)}$ | $v_{(1,2)}$ | … | $v_{(1,n)}$ |
| $v_{(2,1)}$ | $v_{(2,2)}$ | … | $v_{(2,n)}$ |
| | | ⋮ | |
| $v_{(m,1)}$ | $v_{(m,2)}$ | … | $v_{(m,n)}$ |

Row-Color

image representation

# Table as Texts or Images (ACL24)




Text (vanilla-T) vs Image (vanilla-V)

- Conclusions
  - Text is better than Image, except FinQA (math reasoning ability)
  - Best representation
    - Image: colored-row
    - Text: Bracket
  - Input both text and image for VLM not always better than either one.
  - Closed model > open model

| Datasets | T+V | T | V | Metric |
|---|---|---|---|---|
| WikiTQ | 80.0 | 75.0 | 60.0 | Acc |
| TabFact | 64.0 | 71.6 | 60.5 | |
| LogicNLG | 48.0 | 51.5 | 54.1 | |
| FinQA | 61.0 | 43.0 | 57.0 | |
| ToTTo | 42.4 | 43.9 | 44.0 | ROUGE-L |
| E2E | 42.6 | 44.8 | 39.4 | |

Table 19: GPT-4's performance when we pass the text representation (T), image representation (V) and both representation (T+V) to the model.

# TableVQA-Bench(arxiv24)

- Compare
  - VLM QA
  - VLM parse HTML + LLM QA
  - Original HTML + LLM QA

- Conclusions
  - VLM QA needs improve
  - Two phase will short the gap, but still worse than LLM QA

| Input Modality | Model | VWTQ | VWTQ-Syn | VTabFact | FinTabNetQA | Avg. |
|---|---|---|---|---|---|---|
| *Multi-modal Large Language Models (MLLMs)* | | | | | | |
| Vision | GPT-4V [1] | **42.5** | **52.0** | **68.0** | **79.6** | **54.5** |
| | Gemini-ProV [23] | 26.7 | 33.2 | 55.6 | 60.8 | 38.3 |
| | SPHINX-MoE-1k | 27.2 | 33.6 | 61.6 | 36.0 | 35.5 |
| | SPHINX-v2-1k | 25.3 | 28.0 | 66.8 | 31.2 | 33.7 |
| | QWEN-VL-Chat [3] | 19.0 | 23.2 | 60.4 | 29.6 | 28.4 |
| | QWEN-VL [3] | 17.2 | 21.2 | 52.0 | 34.0 | 26.5 |
| | SPHINX-MoE | 15.3 | 16.8 | 58.8 | 2.8 | 20.7 |
| | SPHINX-v1-1k [14] | 13.2 | 17.2 | 58.0 | 3.2 | 19.7 |
| | mPLUG-Owl2 [26] | 10.7 | 14.4 | 56.8 | 2.8 | 17.7 |
| | LLaVA-1.5 [16] | 12.4 | 12.4 | 55.6 | 0.8 | 17.7 |
| | CogVLM-1k [24] | 9.7 | 11.6 | 52.0 | 4.8 | 16.3 |
| | SPHINX-v1 [14] | 7.1 | 9.6 | 55.2 | 1.2 | 14.5 |
| | CogAgent-VQA [7] | 0.3 | 0.8 | 58.4 | 22.8 | 13.8 |
| | InstructBLIP [6] | 5.9 | 6.4 | 50.4 | 0.4 | 12.5 |
| | BLIP-2 [13] | 5.2 | 5.6 | 51.6 | 0.4 | 12.2 |
| | CogVLM [24] | 0.8 | 0.8 | 40.8 | 1.2 | 7.5 |
| | CogAgent-VQA* [7] | 37.2 | 41.2 | 58.4 | 22.8 | 39.0 |
| *Table Structure Reconstruction + Large Language Models (LLMs)* | | | | | | |
| Vision | GPT-4V [1] → GPT-4 [2] | **45.2** | **55.6** | **78.0** | **95.2** | **60.7** |
| | Gemini-ProV → Gemini-Pro [23] | 34.8 | 40.4 | 71.0 | 75.6 | 48.6 |
| *Large Language Models (LLMs)* | | | | | | |
| Text | GPT-4 [2] | **68.1** | **69.6** | **80.0** | **98.8** | **75.5** |
| | Gemini-Pro [23] | 56.4 | 61.2 | 69.6 | 96.4 | 66.1 |
| | GPT-3.5 | 50.5 | 54.4 | 68.0 | 93.2 | 61.2 |
| | Vicuna-13B [5] | 32.8 | 39.2 | 57.6 | 84.8 | 46.7 |
| | Vicuna-7B [5] | 21.5 | 34.4 | 54.0 | 68.8 | 37.0 |

https://arxiv.org/pdf/2404.19205

# Q&A